НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМЕНІ ІГОРЯ СІКОРСЬКОГО»
ФАКУЛЬТЕТ ІНФОРМАТИКИ ТА ОБЧИСЛЮВАЛЬНОЇ ТЕХНІКИ
Кафедра автоматизованих систем обробки інформації і управління

УДК: 004.42

«До захисту допущено»
В.о. завідувача кафедри
_____ *О.А.Павлов*
(підпис) (ініціали, прізвище)
“___” _____2019 р.

Дипломний проект
на здобуття ступеня бакалавра

з напряму підготовки *6.050101 «Комп’ютерні науки»*

на тему: «*Задача забезпечення безпеки ресурсів інформаційної системи на базі графової моделі*»

Виконала: студентка *4* курсу, групи *ІС-51*

Асламова Маргарита Сергіївна _____
(прізвище, ім’я, по батькові) (підпис)

Керівник *доц., к.т.н., доц. Нестеренко О.В.* _____
(посада, науковий ступінь, вчене звання, прізвище та ініціали) (підпис)

Консультант з графічної документації *доц., к.т.н., доц. Тєлишева Т.О.* _____
(посада, науковий ступінь, вчене звання, прізвище та ініціали) (підпис)

Рецензент *провідний науковий співробітник, к.ф.-м.н. Нетесін І.Є.* _____
(посада, науковий ступінь, вчене звання, прізвище та ініціали) (підпис)

Засвідчую, що у цьому дипломному проекті немає запозичень з праць інших авторів без відповідних посилань.
Студентка Асламова М.С. _____
(підпис)

Київ – 2019 року

Національний технічний університет України
“Київський політехнічний інститут імені Ігоря Сікорського”

Факультет (інститут) *інформатики та обчислювальної техніки*
(повна назва)

Кафедра *автоматизованих систем обробки інформації та управління*
(повна назва)

Рівень вищої освіти – перший (бакалаврський)

Напрям підготовки (програма професійного спрямування) *6.050101*

«Комп’ютерні науки» («Інформаційні управляючі системи та технології»)

ЗАТВЕРДЖУЮ
В.о. завідувача кафедри
_____ *О.А. Павлов*
(підпис) (ініціали, прізвище)

“___”_____2019 р.

З А В Д А Н Н Я
НА ДИПЛОМНИЙ ПРОЕКТ СТУДЕНТУ

Асламової Маргарити Сергіївни
(прізвище, ім’я, по батькові)

1. Тема проекту «Графова модель безпеки ресурсів інформаційної системи»

керівник проекту *Нестеренко Олександр Васильович, к.т.н., доцент*
(прізвище, ім’я, по батькові, науковий ступінь, вчене звання)

затверджені наказом по університету від “*23*”*квітня* 2019 р. №1181-с

2. Термін подання студентом проекту “*03*”*червня* 2019 року

3. Вихідні дані до проекту

Технічне завдання

4. Зміст пояснювальної записки

1. Загальні положення: основні визначення та терміни, опис предметного середовища, огляд ринку програмних продуктів, постановка задачі

2. Математичне забезпечення: змістовна та математична постановки задачі, обґрунтування та опис методу розв’язання

3. Програмне та технічне забезпечення: засоби розробки, вимоги до

технічного забезпечення, архітектура програмного забезпечення, побудова звітів

5. Технологічний розділ: керівництво користувача, методика випробувань програмного продукту

5. Перелік графічного матеріалу

1. Схема структурна варіантів використань

2. Схема структурна класів програмного забезпечення

3. Схема структурна послідовності

4. Схема структурна розгортання

5. Креслення вигляду екранних форм

6. Рішення з математичного забезпечення

6. Консультанти розділів проекту

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв

7. Дата видачі завдання *«15» лютого 2019 року*

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів виконання дипломного проекту	Термін виконання етапів проекту	Примітка
1.	*Вивчення рекомендованої літератури*	*17.04.2019*	
2.	*Аналіз існуючих методів розв'язання задачі*	*18.04.2019*	
3.	*Постановка та формалізація задачі*	*19.04.2019*	
4.	*Розробка математичного забезпечення*	*21.042019*	
5.	*Алгоритмізація задачі*	*25.04.2019*	
6.	*Обґрунтування використовуваних технічних засобів*	*26.04.2019*	
7.	*Розробка програмного забезпечення*	*18.05.2019*	
8.	*Налагодження програми*	*19.05.2019*	
9.	*Виконання графічних документів*	*23.05.2019*	
10.	*Оформлення пояснювальної записки*	*29.05.2019*	
11.	*Подання ДП на попередній захист*	*30.05.2019*	
12.	*Подання ДП на основний захист*	*03.06.2019*	
13.	*Подання ДП рецензенту*	*05.06.2019*	

Студент _____ М.С. Асламова
(підпис)

Керівник проекту _____ О.В. Нестеренко
(підпис)

№ з/п	Формат	Позначення	Найменування	Кількість листів	Примітка
			Документи текстові		
1	А4	ДП ІС-5102.1181-с.ПЗ	Пояснювальна записка	75	
2	А4	ДП ІС-5102.1181-с.ТЗ	Технічне завдання	11	
3	А4	ДП ІС-5102.1181-с.ПМВ	Програма та методика випробувань	9	
			Матеріали графічні		
4	А4	ДП ІС-5102.1181-с.ССВ	Схема структурна варіантів використань	1	
5	А4	ДП ІС-5102.1181-с.ССК	Схема структурна класів програмного забезпечення	1	
6	А3	ДП ІС-5102.1181-с.ССП	Схема структурна послідовності	1	
7	А4	ДП ІС-5102.1181-с.ССР	Схема структурна розгортання	1	
8	А3	ДП ІС-5102.1181-с.КЕ	Креслення вигляду екранних форм	1	
9	А3		Рішення з математичного забезпечення	1	

Зм.	Арк.	ПІБ	Підп.	Дата	ДП ІС-5102.1181-с.ВДП		
Розробн.		Асламова М.С.			Відомість дипломного проекту	Літ.	Лист 1 / Листів 1
Керівн		Нестеренко О.В.				КПІ ім. Ігоря Сікорського кафедра АСОІУ гр. ІС-51	
Н/контр.		Тєлишева Т.О					
В.о.зав каф.		Павлов О.А.					

Пояснювальна записка
до дипломного проекту

на тему: Графова модель безпеки ресурсів інформаційної системи

Київ – 2019 року

АНОТАЦІЯ

Структура та обсяг роботи. Пояснювальна записка дипломного проекту складається з п’яти розділів, містить 63 сторінок, 28 рисунків, 4 таблиць, 2 додатки, 14 джерела.

Дипломний проект присвячений побудові графової моделі безпеки ресурсів інформаційної системи.

У розділі загальних положень були розглянуті процеси діяльності, предметне середовище та наявні аналоги програмного продукту.

У розділі інформаційного забезпечення розглянуто вхідні та вихідні дані.

У розділі з програмного забезпечення наведена структура пакетів, їх взаємодія у вигляді послідовності та основні компоненти програми, наведені вимоги до технічного забезпечення.

У технологічному розділі продемонстрована робота програмного продукту.

БЕЗПЕКА, РЕСУРСИ, ЗАГРОЗИ, МЕХАНІЗМИ ЗАХИСТУ, ЙМОВІРНІСТЬ, ІНФОРМАЦІЙНА СИСТЕМА, ЦІННІСТЬ РЕСУРСІВ.

ABSTRACT

Structure and scope of work. The explanatory note of the graduation project consists of five sections, containing 63 pages, 28 pictures, 4 tables, 2 applications, 14 sources.

The graduation project is devoted to the construction of a graph model of the security of resources of the information system.

In the section of the general provisions, the processes of activity, the subject environment, the subject environment and the available analogues of the software product were considered.

In the section of the information support, the input and output data are considered.

In the software section, the structure of the packages, their interaction in sequence and the main components of the program are presented. The requirements for the technical support are specified.

The technology section demonstrates the work of the software product.

SAFETY, RESOURSES, THREATS, PROTECTION MECHANISMS, LIABILITY, INFORMATION SYSTEM, VALUE OF RESOURSES.

Зміст

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

АІС – автоматизована інформаційна система.

АС – автоматизована система.

ГМБ – графова модель безпеки.

ПЗ – програмне забезпечення.

ПК – персональний комп’ютер.

РС – рекомендаційна система.

ВСТУП

Процес дослідження безпеки ресурсів інформаційної системи – це дуже важливий процес пошуку оптимальних алгоритмів для підбору найбільш ефективного захисту системи з урахування різних типів ресурсів та можливих загроз.

У сучасному світі дуже різко зростає кількість користувачів інформаційних систем, щодня збільшується кількість файлів, використовуються дуже великі об’єми даних. Цінність інформації зростає, з’являється багато типів ресурсів і всі вони потребують захисту. Разом з тим зростає і кількість спроб несанкціонованого доступу до інформації та кількість загроз. Саме тому для того щоб запобігти втратам, які можуть виникнути в разі пошкодження інформації, захист ресурсів є найбільш актуальною проблемою.

Сьогодні більшість державних установ та великих компаній прагнуть захистити дані від сторонніх осіб та несанкціонованого доступу. Зараз існує не один механізм захисту системи, саме тому основна задача полягає в тому, щоб підібрати оптимальний варіант, врахувавши вартість його встановлення. Іноді втрати, які можуть виникнути набагато менше ніж вартість системи захисту. Саме тому потрібно ще й визначити доцільність її встановлення. Для цього потрібно провести повний аналіз усіх наявних типів ресурсів, усіх наявних загроз, визначити цінність ресурсів, ймовірності виникнення загроз, а також проаналізувати існуючі механізми захисту. Але треба ще враховувати той фактор, що жоден механізм захисту на дає 100% гарантії.

Враховуючи усе вищесказане виникає необхідність створення такого програмного продукту, який би проводив повний аналіз системи та видавав певні кількісні оцінки, на основі яких можна зробити висновки щодо доцільності встановлення захисту для обраної системи.

Найбільш гарним методом представлення інформації є поєднання графічної та цифрової. Саме тому побудова графової моделі полегшить сприйняття інформації.

1 ЗАГАЛЬНІ ПОЛОЖЕННЯ

1.1 Опис предметного середовища

На кожній стадії розвитку суспільства існує потреба в управлінні інформацією. Саме тому ще з моменту появи людства існують інформаційні системи, які тісно пов'язуються з системами обміну, збереження та отримання інформації. Системи включають в себе низку методів, за допомогою яких здійснюється збирання, зберігання та передача вибраної інформації. Метою створення таких систем є отримання інформації для ефективного управління діяльністю певної організації та створення для цього технічного та інформаційного середовища. З розвитком комп'ютерних технологій суспільство почало використовувати автоматизовані інформаційні системи (АІС), а саме сукупність пов'язаних між собою даних, програмних та апаратних засобів, процедур, що створені для збору, обробки, зберігання та представлення інформації з урахуванням визначених цілей організації.

На даний момент у великих організаціях ефективно використовуються (АІС), що вирішують низку задач. Основними задачами є реалізація інтелектуальної діяльності для створення інформації, її обробка, зберігання та отримання у потрібний час у потрібній формі. При створенні таких інформаційних систем застосовують методи математичної статистики, моделі прогнозно-аналітичних розрахунків та багато інших прикладних засобів.

Створення ефективної (АІС) інформаційної системи потребує поєднання апаратного та програмного забезпечення. Програмне забезпечення відіграє дуже важливу роль у створенні інформаційної системи та включає у себе сукупність усіх необхідних документальних та програмних засобів. Зазвичай розрізняються базове та прикладне програмне забезпечення. Базове ПЗ створює середовище для роботи прикладних програм та повністю забезпечує роботу апаратного забезпечення, а також є найголовнішою частиною для створення ефективної роботи інформаційної системи.

Основною задачею прикладного ПЗ є виконання конкретних задач користувача та організація обчислювальних процесів у системі.

З розвитком інформаційних технологій стрімко збільшуються об'єми даних, що використовують для вирішення великої кількості проблем та завдань. При цьому до цих даних звертається багато користувачів. Це збільшує вразливість інформації, тобто створює можливості для несанкціонованого використання, знищення чи пошкодження. Це відбувається внаслідок доступу до конфіденційних даних користувачів, що не мають спеціальних повноважень. Саме тому розробляють спеціальні механізми захисту інформації, що в свою чергу, зменшує ймовірність несанкціонованого використання.

Основним поняттям інформаційної та кібербезпеки є загроза (Threat), яка може викликати появу нештатних ситуацій у системі. Також до комплексу понять необхідно віднести такі, як порушник (Violator), власник ресурсу (Proprietor of resource), уразливість (Impressionability), засоби захисту (Facilities of defence), ризики (Risks).

Для онтологічного опису сфери кібербезпеки автоматизованої системи (АІС) необхідно визначити властивості захищеності ресурсів, які з найбільшою ймовірністю порушуються внаслідок впливу загроз, тобто здійснити ідентифікацію загроз у вигляді їх переліку із зазначенням відповідності властивостям інформаційних ресурсів, на порушення яких вони спрямовані (порушення конфіденційності, цілісності, доступності та ін.).

У багатьох випадках для АС є необхідність передачі інформації через незахищене середовище (мережа Інтернет).

Саме тому розглянемо загальний випадок класифікації загроз, що можуть виникнути відносно вибраної АС. Для аналізу загроз ресурсам АС необхідним є визначення можливих каналів та видів загроз, що можуть бути реалізовані відносно системи чи інформації, а також аналіз основних джерел їх походження.

Прикладом узагальненої класифікації, що описує існуючі загрози інформаційній безпеці, за якою кожна із загроз підпадає тільки під одну класифікаційну ознаку, і яка тому найбільш застосовна для аналізу ризиків у реальних АС, є класифікація Digital Security Classification of Threats, створена фахівцями компанії Digital Security. Використовуючи цю класифікацію онтологію загроз Реєстру можна представити онтографом, підготовленим програмним засобом Protege на рисунку 1.1. В онтографі кожен вид загроз має унікальне кодування, що забезпечує подальшу формалізацію, пошукові операції та ін.

Аналогічний підхід застосуємо для класифікаційного опису ресурсів АС на рисунку 1.2. Згідно з прийнятою в теорії захисту інформації термінологією ресурси, що захищаються, прийнято називати об’єктами захисту, або просто об’єктами.

Опис характеристик можливих загроз для вибраної АС наведено в таблиці 1.1. Даний опис сформовано виходячи з того, що для реалізації загроз порушник може діяти дистанційно (через засоби зв’язку, витоки інформації, засоби спеціального впливу технічними каналами), чи безпосередньо (в тому числі і шляхом фізичного впливу) на елементи системи. З використанням подібного підходу сформовано й опис ресурсів АС, представлений у таблиці 1.2.

У цих таблицях для спрощення подальшого викладення кодові позначення зведені до скороченого упорядкованого вигляду.

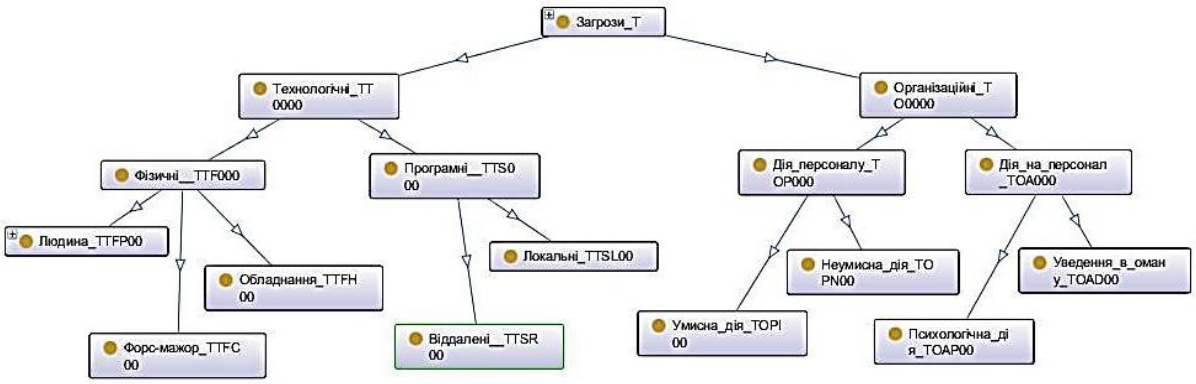


Рисунок 1.1 – Схема структурна «Онтологія загроз»

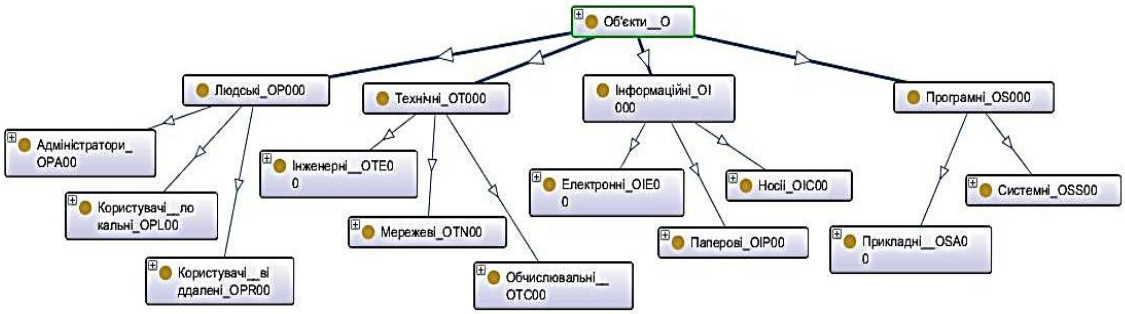


Рисунок 1.2 – Онтологія ресурсів (об'єктів захисту)

Оцінка загрози зазвичай випливає передусім з імовірності її виникнення. Оцінка загроз, наведена в таблиці 1.1, базується на методології нормативних документів і представлена у вигляді якісної оцінки – лінгвістичної змінної, яка приймає значення: незначна, низька, висока, неприпустимо висока – при допущенні, що закон розподілу імовірностей кожної з них є рівномірним, тобто найгіршим для реалізації захисту.

Оцінка рівня шкоди, нанесеної ресурсу внаслідок можливості реалізації загрози розглядається як очікувані збитки від втрати об'єктами захисту кожної з властивостей захищеності. Ця оцінка, наведена в таблиці 1.2, здійснена також за якісною шкалою (низька, середня, висока, неприпустимо висока).

Таблиця 1.1 Характеристика загроз ресурсам АС

Т02	Порушення режимів функціонування (виведення з ладу) систем життєзабезпечення АС (інженерних комунікацій)	низька
Т03	Пошкодження носіїв інформації	висока
Технологічні/фізичні/обладнання		
Т04	Перехоплення даних, що передаються, зміна (модифікація) інформації повідомлень з використанням спеціального обладнання	незначна
Т06	Порушення режимів функціонування АІС шляхом застосування електромагнітного випромінювання	низька
Технологічні/фізичні/форс-мажор		
Т07	Зміна умов фізичного середовища у наслідок стихійного лиха (землетрус, повінь, пожежа, аварії водогону)	незначна
Т08	Збої та відмови у роботі технічних засобів АІС у наслідок аварійного відключення живлення	низька
Т09	Впливи природних електромагнітних завад (грозові розряди, іскріння в електромережах під час електрозварювання та т.ін.)	низька
Технологічні/програмні (локальні та віддалені)		
Т10 Т11	Модифікація програмного забезпечення	низька
Т12 Т13	Впровадження і використання комп'ютерних вірусів	висока
Т14 Т15	Доступ до даних з порушенням встановлених правил розмежування доступу з метою ознайомлення, модифікації, копіювання, знищення даних тощо	низька
Т16 Т17	Отримання захищених даних за допомогою спеціально організованої серії санкціонованих запитів	низька
Т18 Т19	Несанкціоноване змінювання повноваження інших користувачів	низька

Продовження таблиці 1.1

Т20	Видавання власних несанкціонованих запитів за запити операційної системи	низька
Т21 Т22	Неправомірна зміна режимів роботи АС (її окремих компонентів, обладнання, програмних засобів тощо), ініціювання технологічних чи тестуючих процесів, які здатні призвести до незворотних змін у системі	низька
Організаційні/дія персоналу/умисні		
Т23	Одержання атрибутів доступу з наступним їх використанням для маскування під зареєстрованого користувача	низька
Т24	Несанкціоноване копіювання носіїв інформації	дуже висока
Т25	Крадіжки носіїв інформації, виробничих відходів (роздруківок, записів тощо)	висока
Т26	Фальсифікація фактів формування та видачі даних	низька
Т27	Підтвердження отримання від деякого користувача даних, сформованих самим порушником	низька
Т28	Підтвердження передачі якому–небудь користувачеві даних, які не передавалися	низька
Т29	Фальсифікація фактів отримання даних	низька
Т30	Впровадження і використання забороненого політикою безпеки ПЗ або несанкціоноване використання ПЗ, за допомогою якого можна одержати доступ до критичної інформації	висока
Т31	Розкриття змісту даних у каналах зв’язку	висока

Продовження таблиці 1.1

Організаційні/дія персоналу/неумисні		
Т32	Помилки при введені даних в систему, видачі даних за невірними адресами внутрішніх і зовнішніх абонентів тощо	висока
Т33	Невиконання організаційних заходів щодо порядку і правил експлуатації чи використання ресурсів АС, передбачених політикою безпеки, посадовими чи іншими, в тому числі технологічними, інструкціями	низька
Т34	Некомпетентне застосування засобів захисту	низька
Т35	Ненавмисне зараження ПЗ комп'ютерними вірусами	низька
Організаційні/дія на персонал/психологічна дія		
Т36	Використання персоналу АС (шантаж, підкуп) з корисливою метою	низька
Організаційні/дія на персонал/уведення в оману		
Т37	Закладення хибних рішень під час проектування, розробки та модифікації компонентів АС (технічних засобів, технології обробки інформації, ПЗ, засобів захисту, структур даних тощо)	низька

Таблиця 1.2 Об'єкти АС ведення Реєстру

Позначення об'єкту	Об'єкти (відповідно до рисунку 1.2)	Рівень шкоди
Технічні інженерні		
О01	Інженерні комунікації електроживлення	неприпустимо високий
О02	Інженерні комунікації охолодження	неприпустимо високий
О03	Інженерні комунікації водообігу	середній
Технічні обчислювальні		
О04	Сервери	неприпустимо високий
О05	Персональні комп'ютери	високий
О06	Периферійне обладнання	середній

Продовження таблиці 1.2

Технічні мережеві		
О07	Мережеве обладнання локальних мереж (кабельна мережа, комутатори та ін.)	високий
О08	Мережеве обладнання зовнішніх мереж (кабельні лінії, маршрутизатори та ін.)	високий
Програмні системні		
О09	Операційні системи, СКБД	високий
О10	ПЗ комплексу засобів захисту	неприпустимо високий
Програмні прикладні		
О11	Клієнтське ПЗ	низький
Інформаційні електронні		
О12	БД, файли	неприпустимо високий
Інформаційні паперові		
О13	Технічна та експлуатаційна документація	низький
О14	Документація з інформаційної безпеки	середній
Інформаційні носії		
О15	Змінні носії з архівною інформацією	високий
Людські		
О16 О17	Користувачі локальні, віддалені	високий
О18	Адміністратори	неприпустимо високий

Сукупності засобів захисту, що функціонують спільно для виконання певного завдання щодо запобігання небезпеки (криптографічні протоколи, засоби захисту операційних систем і т.ін.) відповідає термін «механізм захисту».

Серед існуючих механізмів і методів захисту виділяють процедурні, програмні та апаратні способи захисту.

На етапі обробки даних використовуються програмні та апаратні способи захисту.

Реалізація процедурних методів полягає у встановленні грифів секретності даних, паролів, створення певних фізичних і організаційних обмежень. А підвищення ефективності даних методів здійснюється за допомогою навчання та підвищення рівня кваліфікації персоналу.

Аналогічно наведемо класифікацію у таблиці 1.3 та онтологічний опис механізмів захисту на рисунку 1.3

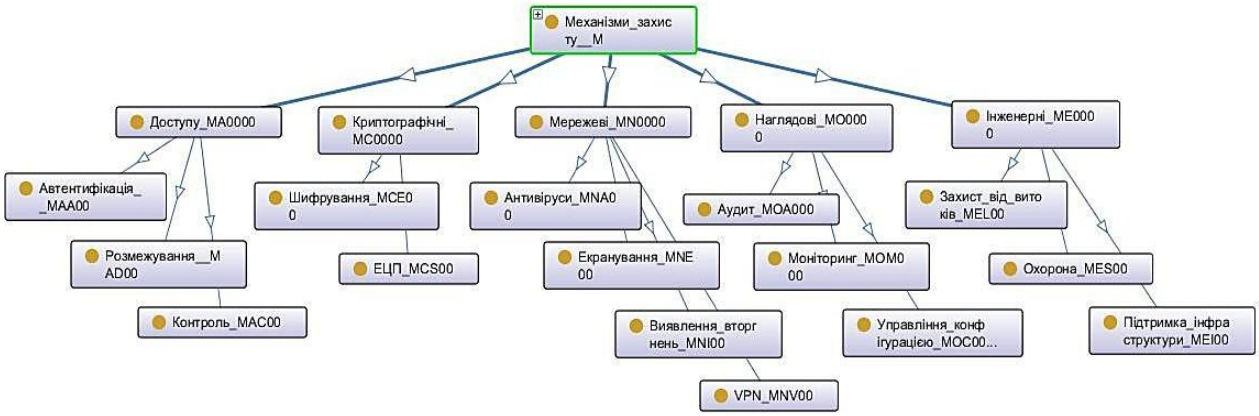


Рисунок 1.3 – Онтологія механізмів захисту АС

Таблиця 1.3 Механізми захисту АС ведення Реєстру

Ім’я вузла графу *M*	Механізми	Вартість реалізації
Доступу		
M01	Автентифікація	незначна
M02	Розмежування доступу	середня
M03	Контроль доступу	висока
Мережеві		
M04	Екранування	значна
M05	Виявлення вторгнень	значна
M06	Антивіруси	незначна
M07	Використання віртуальних приватних мереж	середня

Продовження таблиці 1.3

Криптографічні		
М08	Шифрування	середня
М09	Застосування електронного цифрового підпису (ЕЦП)	незначна
Наглядові		
М10	Моніторинг	середня
М11	Аудит	висока
М12	Управління конфігурацією	значна

В постановках задач для створення систем захисту даних в якості обмежувальних факторів і визначення ефективності постають вартісні і часові затрати на розробку і експлуатацію методів захисту, середній час і ймовірність несанкціонованого доступу до даних, а також втрати від злому системи захисту.

Саме тому інформаційна система потребує підбору оптимальної системи захисту з найменшими витратами та максимальним захистом важливих даних інформаційної системи.

Сама система захисту не приносить жодного прибутку, але при її відсутності є велика ймовірність отримати великі збитки через втрати конфіденційності внаслідок несанкціонованого доступу до інформації або через знищення чи спотворення важливої інформації.

Одним з основних пунктів при підборі оптимальної системи захисту є визначення типів загроз, від яких буде здійснено захист. Також необхідно взяти до уваги і витрати на реалізацію. Це можуть бути і матеріальні затрати на придбання програмного забезпечення та обладнання, а також затрати на шифрування та дешифрування. Витрати не повинні перевищувати можливих збитків внаслідок дії загрози, враховуючи ймовірність їх появи.

Жоден з окремих механізмів захисту не спроможний повністю захистити систему, тому необхідно підібрати комплекс взаємодоповнюючих

компонентів, що зможуть гарантувати адекватну безпеку інформаційної системи.

Збільшення кількості автоматизованих систем, що забезпечують збереження і обробку державних інформаційних ресурсів, наявність в них різних уразливостей та вад захисту впливає на зростання кількості і різноманіття загроз з одночасним зростанням втрат від реалізації цих загроз. Обізнаність власника ресурсу про можливі загрози конкретній системі і пов'язані ризики створює умови для своєчасного застосування відповідних контрзаходів і сприяє зменшенню ризиків. У цьому питанні важливого значення набуває створення адекватної моделі, яка б відображала взаємозв'язок загроз, значення (цінності) ресурсів, очікуваних ризиків та механізмів захисту.

Досі залишаються невирішеними питання, що полягають у теоретичному вивченні та розкритті підходів до опису взаємодії загроз і різних властивостей інформаційних одиниць (об'єктів) для вибору засобів захисту, які необхідні для розв'язання задач забезпечення безпеки. Перспективним вважається представлення моделей у вигляді графів, але й цей підхід потребує подальшого розвитку.

Проблеми побудови моделей безпекового середовища інформаційних ресурсів, визначення впливів різного типу загроз і порушників та забезпечення захищеності ресурсів висвітлено в працях вітчизняних авторів В.Л. Бурячка, В.С. Василенка, В.В. Домарева, А. Б. Качинського, О.Я. Матова, О.М. Новікова, В.О. Устименка, В.О. Хорошка, О.К. Юдіна та інших. Серед іноземних авторів виділяють Edward G. Amoroso, Siri Bromander, Bruce Schneier, Adam Shostack та ін.

Графові моделі є найбільш наочними для моделювання системи захисту та легкими при побудові.

Як бачимо, створення системи для аналізу способів та механізму захисту, визначення оптимального варіанту захисту системи є досить актуальною проблемою на сьогодні.

1.2 Опис процесу діяльності

Складовими частинами інформаційної системи підбору оптимального механізму захисту файлів є:

- підсистема обліку та обробки усіх типів ресурсів;
- підсистема обліку та обробки усіх типів загроз;
- підсистема обліку та обробки механізмів захисту.

Для автоматизації доцільним є створення програмного забезпечення, що аналізує потреби користувача та на основі аналізу видає результат – найкращий варіант захисту необхідної інформації.

Для цього користувач має задати типи файлів для захисту, а також визначити важливість цих файлів. На основі цих даних ПЗ визначає перелік загроз, що мають високу ймовірність здійснення, рахує збитки від настання цих загроз та пропонує механізми захисту цих файлів. Таким чином за допомогою цього ПЗ користувач зможе ,в залежності від своєї мети, оцінити необхідність захисту та визначити оптимальний захист системи щоб в майбутньому не зазнати значних втрат.

На сьогоднішній день захист інформації є дуже актуальною проблемою і державні установи, великі компанії прагнуть захистити дані від сторонніх осіб та несанкціонованого доступу. Саме тому створення такого ПЗ є досить актуальним.

1.3 Опис функціональної моделі

Механізм роботи будь-якої рекомендаційної системи (РС), як правило, включає три складові:

- довідкові дані;

- вхідні дані;

- алгоритм.

Довідковими даними є типи файлів для захисту, класифікація загроз, існуючі механізми захисту для кожного типу загроз.

Вхідні дані – це інформація про типи ресурсів, загроз, наявні механізми систем захисту, а також кількісні оцінки.

Алгоритм як метод взаємодії двох вищеописаних складових слугуватиме для аналізу та підбору оптимальної системи захисту.

Для визначення логіки системи використовуємо діаграму варіантів використання (use-case diagram). Її призначенням є побудувати модель того, як повинна функціонувати система.

Ключовими елементами моделі діаграми варіантів використання є:

- актор, який позначає ролі користувача, що взаємодіє з деякою сутністю;
- прецедент, що відображає дії, які виконуються в системі та дають бажані результати акторам.

Система підбору оптимального механізму захисту, що розробляється в рамках дослідження, матиме двох акторів:

- користувач;
- адміністратор.

Прецедентами відображені основні функції бізнес-процесу, котрий розглядається.

По суті, користувач обирає типи файлів, які потребують захисту, а також визначає їх важливість. Аналіз введених даних та рекомендація оптимального механізму захисту, здійснюється системою.

Акторів та їх головні функції наведено на діаграмі прецедентів UML в графічному матеріалі.

Подальший опис функціональної моделі будемо здійснювати за допомогою діаграми IDEF0.

Широко використовується методологія IDEF0 завдяки своїй простій і зрозумілій графічній нотації, застосування якої є дуже зручним для побудови моделі. Головне місце в методології відводиться діаграмам. На діаграмах відображаються зв'язки між функціями та зовнішнім середовищем, функції системи за допомогою геометричних прямокутників. Зв'язки відображаються за допомогою стрілок.

У IDEF0 реалізовані три базові принципи моделювання процесів:

- принцип функціональної декомпозиції являє собою спосіб моделювання типової ситуації, коли будь-яка дія, операція, функція можуть бути розбиті на більш прості дії, операції, функції. На рисунках 1.5 – 1.6 показана декомпозиція бізнес – функцій, які представлені у вигляді сукупності елементарних функцій;

- принцип обмеження складності. При роботі з IDEF0 діаграмами істотним є умова їх легкості і розбірливості читання. Суть принципу обмеження складності полягає в тому, що кількість блоків на діаграмі має бути в діапазоні від двох до шести на рисунках 1.4-1.5. Практика показує, що дотримання цього принципу призводить до того, що функціональні процеси, представлені у вигляді IDEF0 моделі, добре структуровані, зрозумілі і легко піддаються аналізу;

- принцип контекстної діаграми. Моделювання ділового процесу починається з побудови контекстної діаграми на рисунку 1.4. На цій діаграмі відображається лише один блок – головна бізнес-функція нашої інформаційної системи, а саме обробка запиту від користувача.

Рисунок 1.4 – Схема структурна контекстної моделі системи

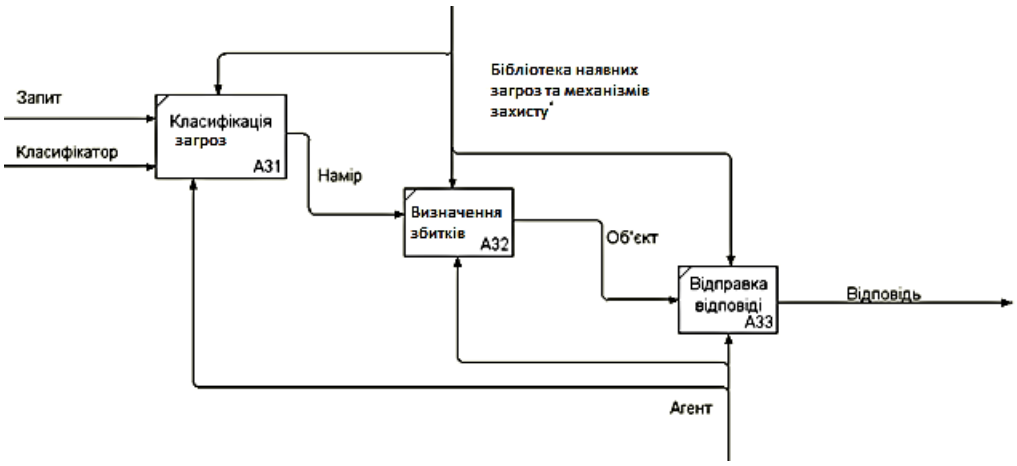


Рисунок 1.5 – Схема структурна функціональної моделі IDEF0

Відповідно до заявленої функціональної моделі, сформулюємо перелік функціональних вимог до системи.

До розроблюваної системи висуваються такі вимоги, як функціональні та нефункціональні. Функціональні вимоги в свою чергу поділяються на вимоги користувачів та системні вимоги.

Таблиця 1.4 Функціональні вимоги

Функціональні вимоги	Пріоритет
1. Система надає можливість користувачу ввести запит 1.1. Система надає можливість переглянути відповідь на запит	Високий Високий
2. Система надає можливість користувачу вибрати типи файлів для захисту 2.1. Система надає можливість переглянути список загроз для обраних файлів	Високий Високий
3. Система надає можливість переглянути усі наявні механізми захисту 3.1. Система надає можливість переглянути механізми захисту для загроз, що можуть виникнути при заданих типах файлів	Високий Високий

Нефункціональні вимоги до системи:

- система має за короткий, обмежений (30 секунд) час видати відповідь на запит користувача;
- система має мати інтуїтивно зрозумілий інтерфейс;
- система має давати доброзичливі відповіді та рекомендації.

1.4 Огляд наявних аналогів

На сьогоднішній день проблема захисту інформації є однією з найважливіших у світі інформаційних технологій. Систем, які б аналізували наявні системи захисту та підбирало б оптимальний варіант захисту системи, розроблено дуже мало. Тому створення системи ГМБ є актуальним рішенням.

1.5 Формулювання призначення і цілей створення системи

Призначення розробки є визначення оптимального набору механізмів захисту, яке необхідно встановити для оптимального захисту інформації від несанкціонованого доступу та пошкодження.

Цілями розробки є:

- полегшити вибір великих компаній та державних установ для захисту та збереження інформації;
- зекономити час та кошти, які були б витрачені на консультацію фахівця з підбору оптимального механізму захисту.

Для досягнення поставлених цілей необхідно налагодити процес швидкого підбору оптимального набору ПЗ для захисту даних в залежності від мети користувача, для цього необхідно:

- створити перелік основних загроз для кожного типу файлів та визначити збитки, які виникнуть внаслідок спрацювання загрози, а також визначити ймовірність настання кожної загрози;
- створити перелік усіх наявних механізмів захисту для кожного типу загроз;
- врахувати важливість інформації, яка подається для захисту;
- на основі аналізу даних визначити кількісні оцінки доцільності використання наявних механізмів захисту.

Висновок до розділу

Отже, був проведений опис предметного середовища, визначені вимоги та аналоги, а також було визначено призначення розробки, цілі та задачі для досягнення цілей. Виходячи з вищевикладеного створення такого застосунку є дуже актуальним.

2 ІНФОРМАЦІЙНЕ ЗАБЕЗПЕЧЕННЯ

2.1 Вхідні дані

Вхідними даними є визначені типи ресурсів та кількісна оцінка їх важливості, визначені типи загроз, ймовірність їх виникнення та їх взаємозв’язок з ресурсами, а також наявні механізми системи захисту та ймовірність знищення ними загроз.

2.2 Вихідні дані

Вихідними даними є кількісні оцінки доцільності використання наявних механізмів захисту.

2.3 Опис структури бази даних

В даному проекті використання бази даних не передбачено. Таблиці з даними о ресурсах, загрозах та механізмах захисту описані у вигляді текстового файлу.

Висновок до розділу

У даному розділі описані вхідні та вихідні дані комплексу задач графової моделі безпеки ресурсів інформаційної системи.

3 МАТЕМАТИЧНЕ ЗАБЕЗПЕЧЕННЯ

3.1 Математична постановка задачі

Специфіку роботи програми представимо у вигляді графу процесів (позначимо його *ГП*) на рисунку 3.1. Вершинами цього графу є основні поняття середовища, а дугами – відношення між ними. Тоді граф *ГП* описує такі процеси. Власник ресурсу (PR) оцінює (1) наявні у нього інформаційні ресурси (IR) з метою визначення їх властивостей, а саме цінності для організації чи компанії. Порушник (V) створює (2) загрозу (T), яка породжує (3) ризик (R) втрати (4) ресурсом (IR) власних властивостей. Водночас загроза дістає (5) вираз у вигляді атаки (A), яка, використовуючи (6) уразливості (I) системи досягає негативних наслідків своєї активності. Між атаками порушник постійно здійснює пошук (8) нових уразливостей в системі, які можуть з'являтися (9) упродовж життєвого циклу ресурсу.

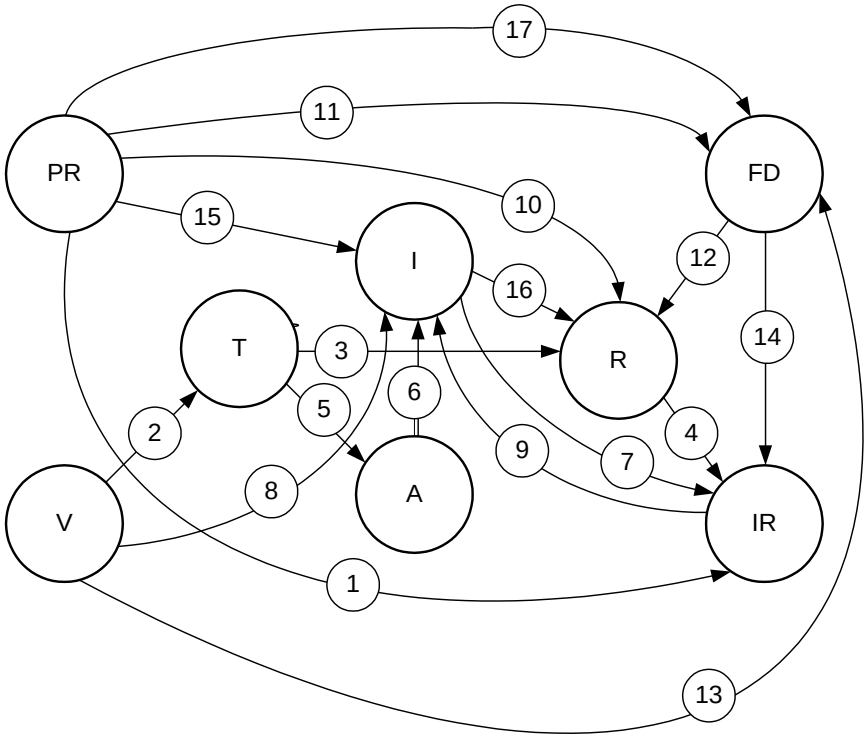


Рисунок 3.1 - Граф *ГП* процесів, що відбуваються в умовах наявності загроз для інформаційних ресурсів

Власник ресурсу, знаючи (10) про наявність ризику втрати ресурсом своїх властивостей створює (11) засоби захисту (FD), які спроможні зменшити (12) ризики. Але засоби захисту також можуть мати свої вади, тому порушник їх знаходить і використовує (13) для доступу (14) до ресурсу. У цих обставинах власник вимушений здійснювати постійний моніторинг (15) уразливостей системи з метою їх своєчасного виявлення і тим самим зменшення (16) ризиків. Водночас необхідним є вживання заходів щодо оцінювання (17) стану засобів захисту з метою виявлення і усунення вад захисту.

Модель відношень множини загроз ***T*** і множини об'єктів ***O*** можна представити дводольним графом *GTO = (V(T,O), E(T,O))* на рисунку 3.2, у якому множини вершин його долей $T \cup O = V(T, O)$, $\mathrm{T} = \{\mathrm{T01}, \mathrm{T02}, \ldots, \mathrm{T37}\}$, ***O*** = $\{\mathrm{O01}, \mathrm{O02}, \ldots, \mathrm{O18}\}$, $T \cap O = \emptyset$ та множина ребер *E(T,O)*, в якому ребро $(T_p, O_q) \in E(T, O)$, якщо є загроза *Tp* об'єкту *Oq*.

У загальному випадку з моделі загроз витікає необхідність захисту від впливу загроз усім властивостям захищеності інформації, насамперед від загроз, наслідком реалізації яких може бути неприпустимо високий чи високий рівень шкоди, оскільки такі загрози мають комплексний, тобто одночасний вплив на декілька властивостей захищеності. Такі загрози прийнято називати найбільш суттєвими (найбільш небезпечними, найбільш ймовірними) загрозами.

Виявлення найбільш суттєвих загроз та високого рівня шкоди, нанесеної ресурсу, є основою для визначення в подальшому потрібних засобів захисту, а отже визначення складу потрібних контрзаходів для забезпечення припустимої захищеності, необхідних для захисту засобів, підсистем, механізмів та функцій захисту, тобто дає змогу будувати відповідні моделі систем захисту.

У відповідності до відомої моделі безпеки з повним перекриттям [10], яка будується виходячи з положення, що система безпеки повинна мати

принаймні один засіб для забезпечення безпеки на кожному можливому шляху дії загрози на об'єкт, в нашій моделі з'являється третій набір, що описує механізми забезпечення безпеки.

$$\boldsymbol{M} = \{M_1,\ M_2, M_3,\ \ldots, M_r\}, r = 1, 2, \ldots.$$

В ідеальному випадку кожен механізм M_k повинен усувати деяке ребро (T_p, O_q). На практиці ж M_k виконує функцію "бар'єру", забезпечуючи деяку міру опору спробам реалізації загрози. Включення в модель множини $\boldsymbol{M}$ перетворює граф *GTO* в тридольний граф

$$GTMO = (V(T, M, O), E(T, M), E(M, O)).$$

Побудова графу $GTMO$ є нетривіальною задачею, зважаючи на складність зв'язків графу *GTO* – навіть для такого не дуже складного прикладу, що розглядається.

Щоб полегшити розв'язання задачі розшукаємо на графі *GTA* компоненти зв'язності, тобто такі його підграфи $G_i = (V_i, E_i)$, що $GTA = \cup Gi$, але $V i \cap V j = \varnothing$ та $Ei \cap Ej = \varnothing$, $i, j = 1,2,\ldots, i \neq j$, у той час як у будь-якому *Gi* будь-які вершини *u* та *v* з'єднані простим ланцюгом. Стрілочки на ребрах графу *GTO* вказують лише на те, що небезпека йде з боку загроз до об'єктів і виконують чисто ілюстративну функцію. Тому можна розглядати граф *G = (V,E)* як неорієнтований.

Для знаходження компонент зв'язності можливо застосувати відомі алгоритми. Більшість алгоритмів на графах використовують їх представлення за допомогою матриці суміжності або списків суміжних вершин. Матриця суміжності *D* позначеного графу з *n* вершинами є матрицею порядку (*n* x *n*), в якій її елемент $x_{ij}=1$, якщо вершина u_i з'єднана ребром з вершиною v_j, в іншому разі $x_{ij}=0$. При заданні графу списком суміжних вершин для кожної вершини задається список вершин, з'єднаних з нею ребрами.

У разі представлення графу за допомогою списків суміжних вершин для пошуку компонент зв'язності зазвичай застосовують алгоритми, які базуються на алгоритмах пошуку в глибину та пошуку в ширину, які

досліджують граф методом обходу усіх вершини і ребер, використовуючи механізми рекурсії, фарбування вершин або ребер, поняття предків і нащадків, міток часу тощо [11,12].

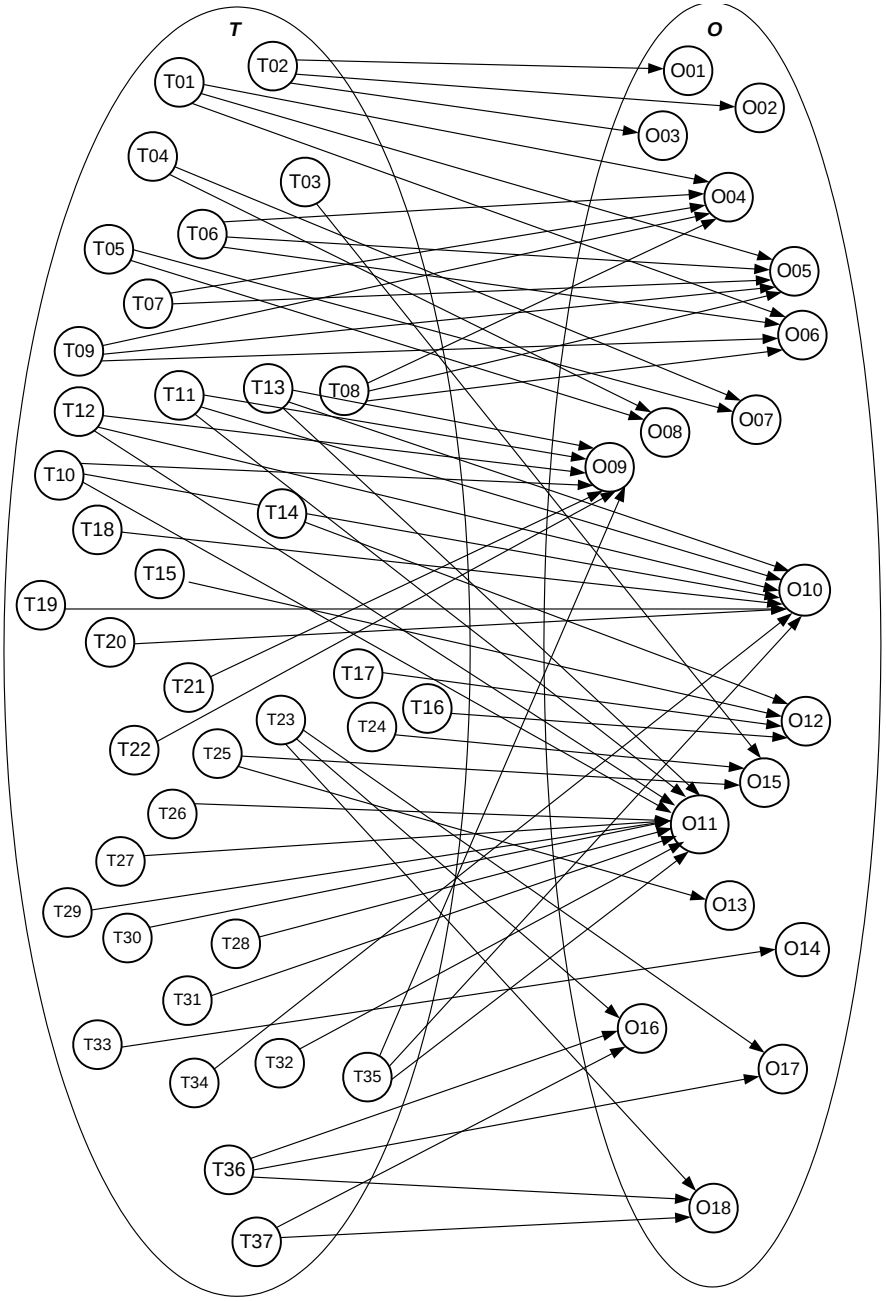


Рисунок 3.2 - Граф *GTO* відношень загроз і ресурсів АС

Кількість операцій для пошуку в глибину або в ширину, а також для пошуку компонент зв'язності, який базується на цих алгоритмах, пропорційна кількості вершин та ребер, взятих разом, тому складає $O(|W|+|E|)$, де константа, схована в позначенні «*O*-велике» не перевищує

декілька десятків. |V| і |E| - кількість відповідно вершин і ребер у множинах V і E.

Стандартний опис нашого графа *GTO* за допомогою списків суміжних вершин виглядає наступним чином:

T01: O04, O05, O06;

T02: O01, O02, O03;

T03: O15;

...

T37: O16, O18;

01: T02;

…

O10: T13, T11, T12, T10, T18, T19, T20;

…

O18: T23, T36, T37.

У разі задання довільного графу *G* матрицею суміжності *D* можливо застосувати алгоритми, які перетворюють її у блочно-діагональну матрицю *B* виду:

$$\begin{pmatrix} B_{12} & \ddots & 0 \\ & B_{ij} & \\ 0 & & \ddots \end{pmatrix}$$

де $B_{12},\ldots,\ B_{ij},\ldots$ є матриці суміжності, які відповідають окремим компонентам зв'язності графу G.

Це досягається, наприклад, шляхом множення матриці D на деяку матрицю перестановки P і її обернену матрицю P^{-1}, тобто $B=P^{-1}DP$. Іншим способом отримання з матриці суміжності *D* блочно-діагональної матриці B є застосування алгоритму, який використовує вектор покажчиків, який містить послідовність перестановок рядків та стовпців матриці *D*.

Кількість операцій в алгоритмах, які використовують матричне представлення, складає $O(|V|^2)$. Перехід від одного представлення до другого може бути виконаний також за $O(|V|^2)$ операцій .

Застосовуючи один із наведених алгоритмів отримуємо шукані компоненти зв'язності $G_i = (V_i, E_i)$ графу *G(V, E)* на рисунку 3.3.

Тепер розглядаючи наведені підграфи $G_i = (V_i, E_i)$ визначення необхідних механізмів захисту з набору *M* вочевидь значно спрощується.

Але залишається проблема оцінки вартості реалізації вибраних механізмів захисту та пріоритетності їх реалізації. Для цього традиційно відомим є визначення передусім оцінки ризиків, яка випливає з ймовірності здійснення загрози та рівня шкоди (збитків) від порушень по кожному з видів порушень.

В умовах, коли кожен ресурс може підпадати під дію кількох загроз, оцінку ризиків доцільно розраховувати у кількісному виразі:

$$S_k = \max_i (P_i \cdot L_{i,k})$$

де S_k- оцінка ризику *k*-го ресурсу;

P_i – ймовірність реалізації *i*–ї загрози;

$L_{i,k}$ – цінність *k*-ого ресурсу.

Для проведення розрахунків якісні оцінки ймовірностей виникнення загроз необхідно представити у кількісному вимірі, наприклад, за такою схемою: незначна - 0.1, низька - 0.3, висока - 0.7, дуже висока - 0.9. Оцінку цінності та вартості зазвичай виражають у грошовій формі. Для попередніх оцінок, зокрема експертних, можна використати бальну оцінку за такою, наприклад, схемою: низька – 20 балів, середня - 50, висока - 70, неприпустимо висока – 90.

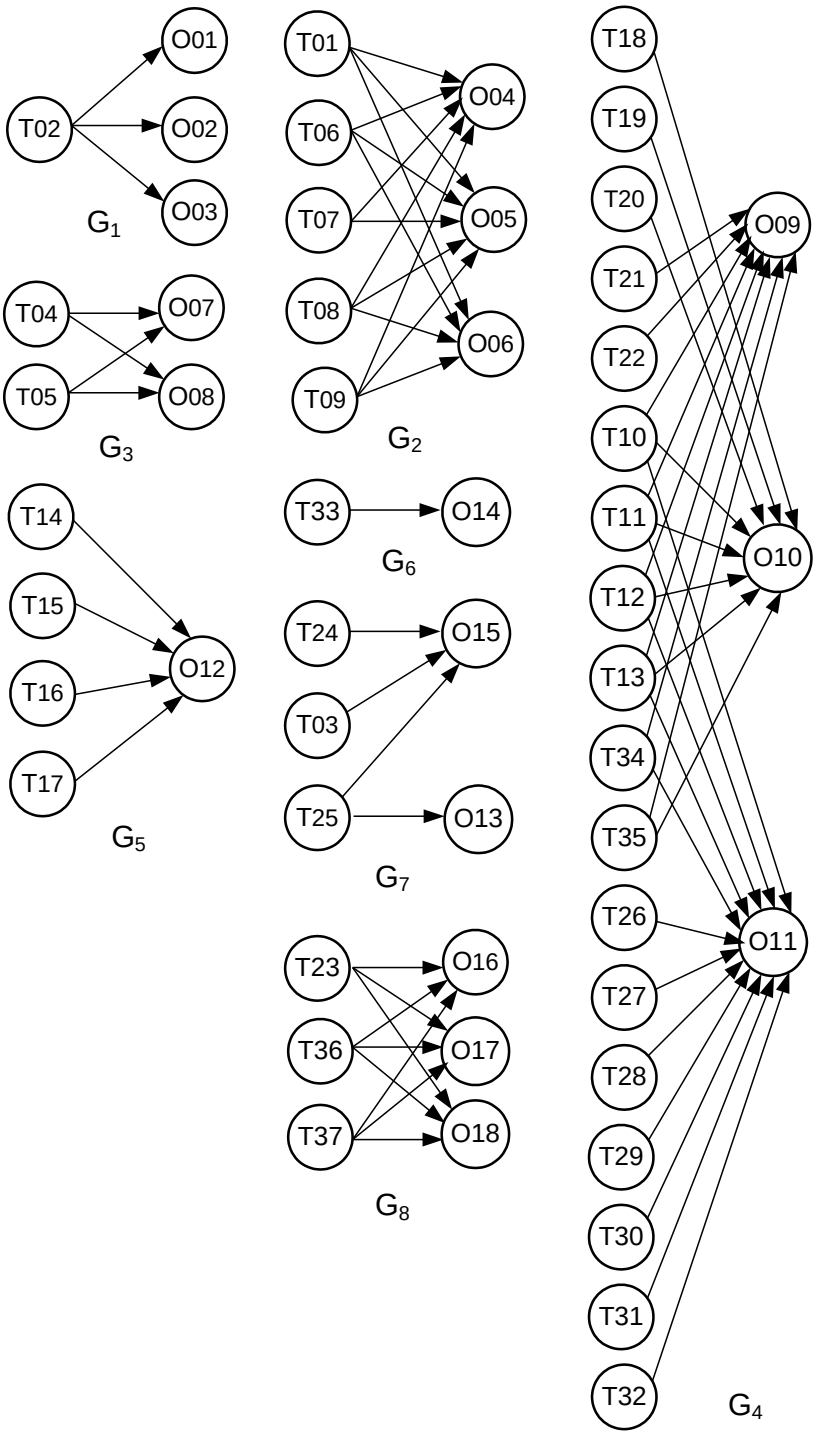


Рисунок 3.3 - Підграфи $G_i = (V_i, E_i)$ графу $G = (V,E)$

Оцінка загрози зазвичай випливає передусім з імовірності її виникнення. Ймовірність виникнення загрози та ймовірність знищення загрози механізмом захисту задається у вхідних даних та є дійсним числом від 0 до 1.

Оцінка цінності ресурсів та вартості встановлення механізмів захисту також задається на вході роботи програми та оцінюється цілим числом за 100-бальною шкалою.

Оцінка рівня шкоди, нанесеної ресурсу внаслідок можливості реалізації загрози розглядається як очікувані збитки від втрати об’єктами захисту кожної з властивостей захищеності.

Для оцінки захисту деякого об’єкту побудуємо тридольний граф на рисунку 3.4.

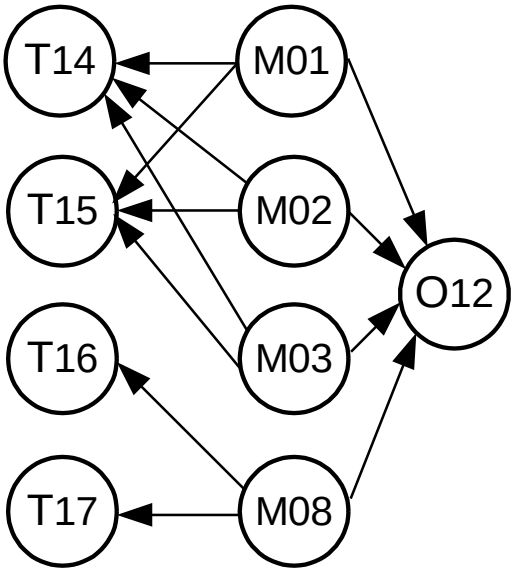


Рисунок 3.4 - Тридольний граф з урахуванням механізмів безпеки для окремого об’єкту

Отже, виділивши тридольні та дводольні підграфи ми маємо змогу проаналізувати та наочно побачити уразливість ресурсів, можемо оцінити втрати від здійснення загроз, а також побачити чи є доцільним встановлення захисту для певного типу ресурсів. На основі вищевикладеного матеріалу маємо конкретну задачу для вирішення, зміст якої наведемо нижче.

Є деяка автоматизована система (АС). Вона складається з множини ресурсів (об’єктів - О), для яких визначена цінність за 100-бальною шкалою. Вони порушуються внаслідок впливу множини загроз Т, які характеризуються імовірністю виникнення. Необхідно визначити механізми захисту М об’єктів О від впливу усіх загроз Т за критерієм мінімуму витрат на механізми і мінімізації ризиків втрати ресурсів.

Для вирішення поставленої задачі будемо дотримуватись такого алгоритму:

КРОК1. Модель відношень множини загроз ***T*** і множини об'єктів ***O*** представимо дводольним графом G_{TO}. Спрямованість загроз *Tp* до конкретного об'єкту *Oq* задається як вхідні дані. Граф G_{TO} будується вручну (з використанням готових засобів - Grafviz).

КРОК2. Формуємо матрицю суміжності D графу G_{TO}.

КРОК3. За допомогою створеної програми знаходимо компоненти зв'язності (підграфи). Алгоритм знаходження підграфів: По матриці D вибираємо об'єкт, на який спрямовано більшість загроз. Добираємо об'єкти, на які спрямовані ці загрози. До них добираємо інші загрози, що на них спрямовані. Далі повторюємо з наступним об'єктом.

КРОК4. Формуємо тридольні підграфи з механізмами захисту за аналогією з КРОК1 Припущення – один механізм може бути бар'єром для кількох загроз.

КРОК5. Формуємо матриці суміжності D_i для кожного підграфу .

КРОК6. За допомогою створеної програми для кожного об'єкту проводимо розрахунки ризиків (формула суми S_k).

КРОК7. Розраховуємо оцінки механізмів захисту $C_м$ шляхом ділення його вартості на кількість загроз, для яких він вибраний бар'єром.

КРОК8. Для кожного об'єкту отримуємо загальну вартість захисту $B_{зо}$, підсумовуючи $C_м$

КРОК9. Для кожного об'єкту порівнюємо $B_{зо}$ та S_k, та визначаємо суттєвість перевищення або недостатність оцінок. (Нормування даних у нас проводиться шляхом переведення в бали)

КРОК10. Робимо висновки: Чим більше $B_{зо}$ над S_k тим краще захищений ресурс. Менше – треба добавляти. В ідеалі перевищення не повинне бути значним (витрачаємо зайві кошти). Це може враховуватись при проектуванні КЗЗ з метою можливого зменшення

його вартості, що може досягатися шляхом інтеграції розрізнених механізмів захисту створенням комплексного бар'єру, що поєднує функціонал зазначених механізмів, і природно може бути дешевше.

Висновок до розділу

В даному розділі була сформульована змістовна та математична постановка задачі побудови графової моделі безпеки ресурсів інформаційної системи, визначено основні кроки роботи алгоритму для визначення кількісних оцінок та проведення аналізу.

4 ПРОГРАМНЕ ТА ТЕХНІЧНЕ ЗАБЕЗПЕЧЕННЯ

4.1 Засоби розробки

У таблиці 4.1 перераховано основні засоби, що були використані при розробці.

Таблиця 4.1 – Основні ресурси при розробці

Операційна система	Windows 10
Мови програмування	C#, ASP.NET
Середовища розробки	Visual Studio
Додаткові засоби	GraphViz
Засоби проектування	Visio, PowerDesigner

Windows 10, як відомо, є операційною системою корпорації Microsoft сімейства Windows NT. Це досить відома та потужна серед звичайних користувачів персональних комп'ютерів операційна система, яка, по суті, стала такою популярною завдяки тому, що на зламі тисячоліть більшість ПК продавалися саме з цією ОС. Більшість програмних продуктів розроблено під операційні системи сімейства Windows, а отже використання даного засобу при розробці збільшує кількість користувачів програмним продуктом, а також підвищує зручність при користуванні та виключає необхідність встановлення додаткової операційної системи.

Програмний продукт розроблений на платформі ASP .NET MVC, яка представляє собою фреймворк для створення сайтів і веб-застосунків за допомогою реалізації паттерна MVC на рисунку 4.1.

Концепція паттерна MVC (model – view - controller) передбачає розділення застосунку на 3 компонента:

Контроллер представляє клас, який забезпечує зв'язок між користувачем і системою, представленням і сховищем даних;

Представлення – це власне візуальна частина програми або користувацький інтерфейс застосунку (як правило, це html- сторінки);

Модель – представляє клас, який описує логіку даних, що використовуються.

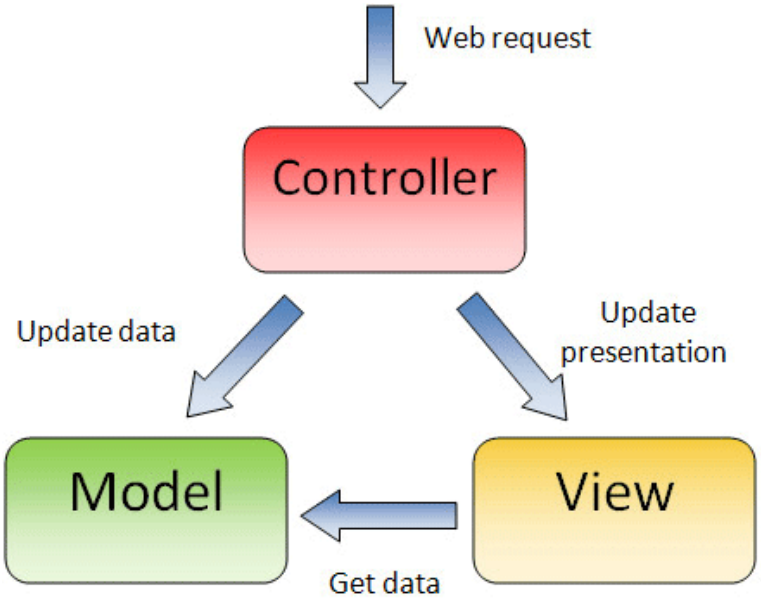


Рисунок 4.1 – Схема, що відображає принцип роботи ASP.NET MVC

C#, являється однією з найпопулярніших мов програмування, тому що вона легка у вивчені, дуже універсальна, має велику кількість модулів та бібліотек.

Алгоритм роботи програми та усі наявні функції написано мовою C#, яка підтримує велику кількість інструментів та фрейморків і є досить популярною й зручною на сучасному ринку розробки систем. Великою перевагою цієї мови, порівняно з іншими мовами програмування, є її гнучкість.

Використовуючи середовище Visual Studio, можливості C# вражають, адже з її допомогою можна створювати додатки для Windows, мобільні додатки, веб-застосування, ігри, програми для Android та iOS, котрі розробляються з використанням додаткових фреймворків.

Visual Studio, як одне із середовищ розробки з великим набором інструментальних засобів, є досить ефективним рішенням створення систем

зі зручним інтерфейсом. Крім того, саме середовище має комфортний інтерфейс на рисунку 4.2.

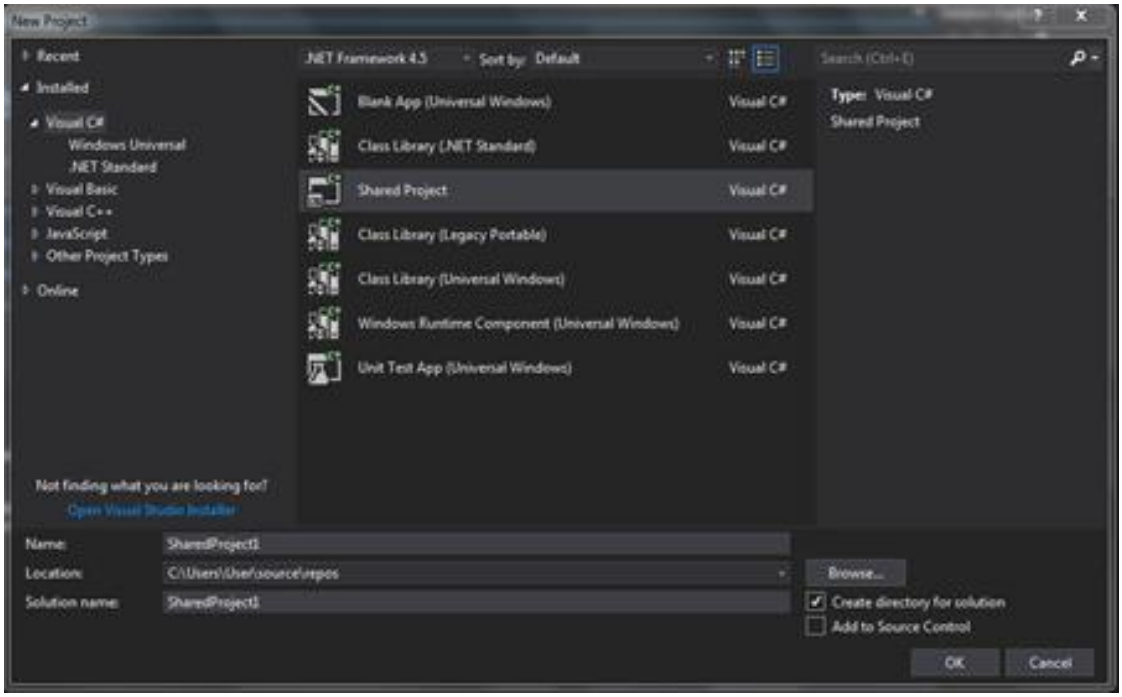


Рисунок 4.2 – Інтерфейс середовища Visual Studio

В якості додаткового програмного забезпечення ми використовували GraphViz для візуального представлення графів. Даний програмний продукт дуже гарно представляє структурну інформацію у вигляді графів, схем для більш наочного сприйняття. Основними перевагами для вибору даного ПЗ було те, що GraphViz використовує дуже простий текстовий опис графів для побудови, а також є можливість зберегти намальований граф у різних форматах. Крім того програма містить дуже багато опцій для змінення дизайну та зовнішнього вигляду графу чи діаграми, а саме вибір кольору, шрифту, а також стилю ліній. Саме тому це є досить зручним та простим у вивченні засобом для графічного представлення інформації. А також GraphViz має зручний інтерфейс на рисунку 4.3.

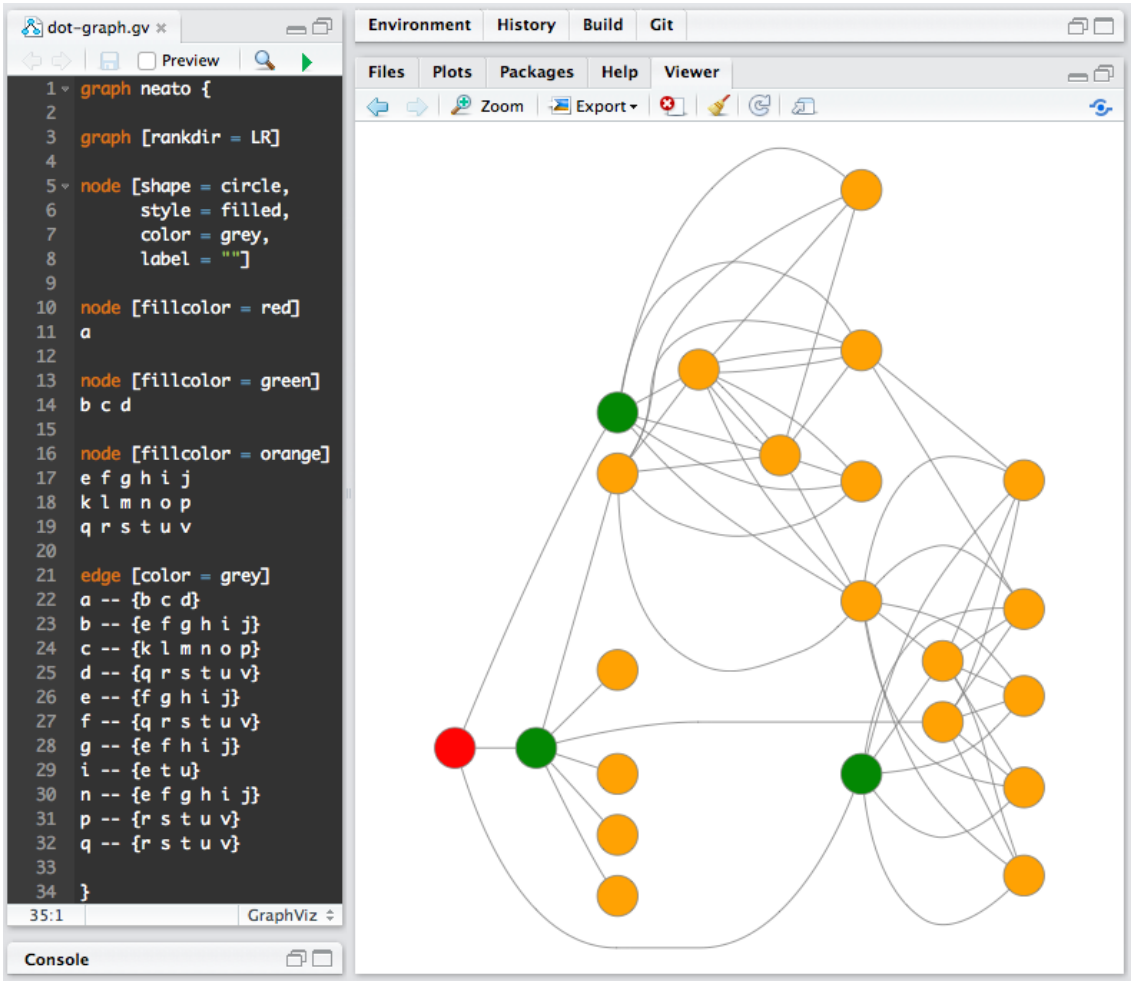


Рисунок 4.3 – Інтерфейс програми GraphViz

Проектування діаграм, бізнес-процесів, розробка схем алгоритмів і т. д. виконувались різних середовищах, таких як Microsoft Visio та PowerDesigner

Visio та PowerDesigner є відомими, досить зручними, графічними редакторами діаграм та блок-схем.

MS Visio являє собою досить простий та гнучкий графічний редактор з комфортним інтерфейсом для швидкої побудови бізнес-моделей та схем невеликої системи. Основними перевагами є легкість у вивченні та швидке освоєння при роботі з програмою, а також оперативна побудова бізнес-моделей та можливість внесення необхідних змін.

PowerDesigner – один з найпопулярніших засобів для управління та створення моделей даних, інформаційної архітектури. Має більше можливостей для побудови діаграм та бізнес-моделей та зручний гнучкий інтерфейс.

4.2 Вимоги до технічного забезпечення

Для правильної роботи даного web застосування до складу технічних засобів повинні входити:

- комп’ютер з такою конфігурацією:
  1) процесор з тактовою частотою не нижче 1 ГГц;
  2) достатній об’єм оперативної пам’яті (не менше 256 МБ);
  3) інші складові можуть мати будь-які параметри, тому що вони не значним чином впливають на роботу програми;
- додатково має бути встановлене таке програмне забезпечення:
  1) операційна система Windows 10;
  2) Net Framework 3.5 і вище;
- комп’ютерна периферія, до складу якої входить:
  1) монітор;
  2) мишка;
  3) клавіатура

4.3 Схема архітектури ПЗ

Наведемо схему архітектури ПЗ на рисунку 4.4.

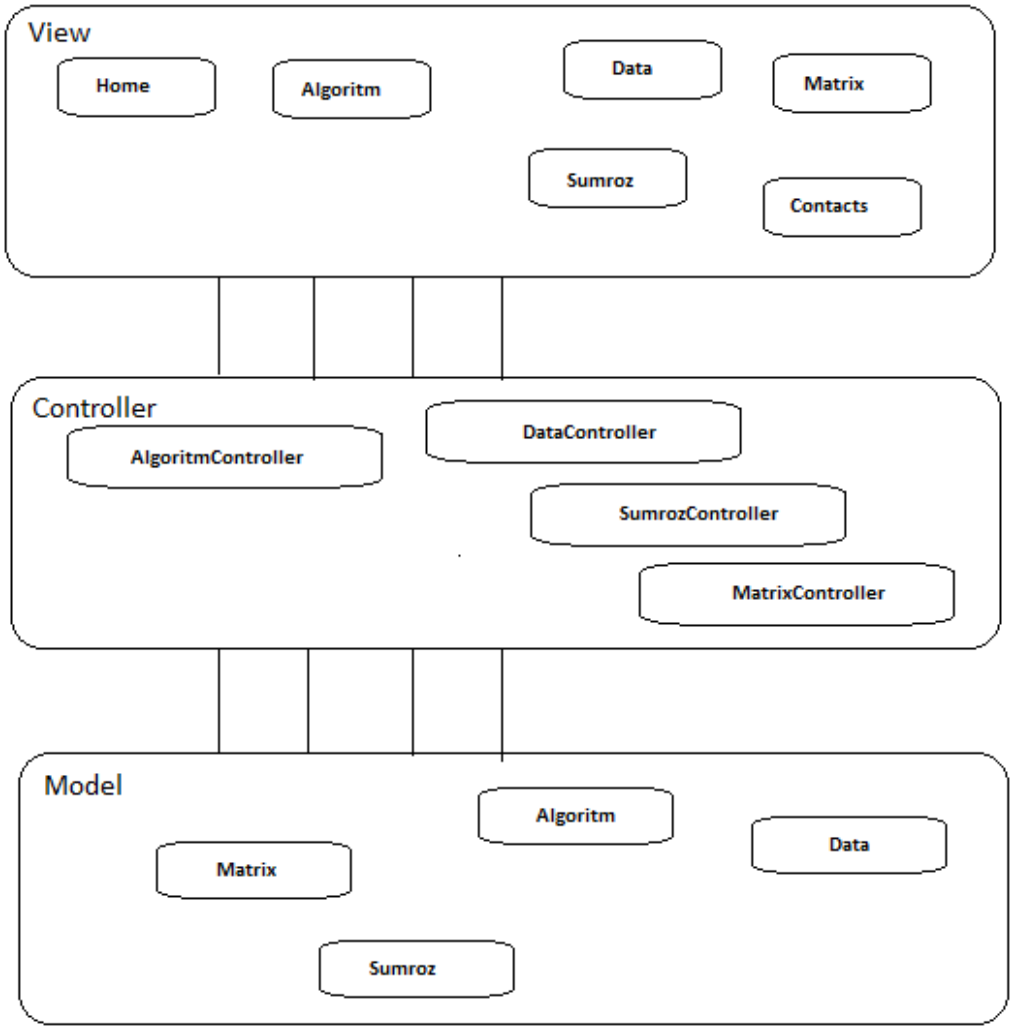


Рисунок 4.4 – Схема структурна архітектури ПЗ

Controllers:

AlgoritmController – контролер для визначення дводольних підграфів та розрахунку кількісних оцінок;

MatrixController – для побудови матриць суміжності, що необхідні для роботи алгоритмів;

SumrozController – для визначення кількісних оцінок ймовірності настання загрози та ймовірності її подолання, цінності ресурсів та вартості механізму захисту;

DataController – для визначення інформації про систему, яка потребує захисту (робота з вхідними та вихідними даними);

Models:

Algoritm.cs – містить реалізацію алгоритму виділення підграфів та розрахунок кількісних оцінок;

Matrix.cs – містить реалізацію алгоритму побудови основних матриць суміжностей для роботи алгоритмів;

Sumroz.cs – містить реалізацію побудови таблиць з кількісними оцінками, а саме ймовірності настання загрози, ймовірності її подолання певним механізмом захисту, а також цінності ресурсів та вартості встановлення кожного механізму захисту;

Data.cs – містить моделі вхідних та вихідних даних;

Views:

Algoritm – сторінка для відображення виділених підграфів та розрахованих кількісних оцінок для аналізу;

Matrix– сторінка для відображення основних матриць суміжностей, що використовуються в роботі алгоритмів;

Data –сторінка для визначення інформації про систему, яка потребує захисту;

Home – головна сторінка;

Contacts – сторінка, що містить інформацію про розробника;

4.4 Діаграма класів

Діаграма класів наведена в частині графічного матеріалу.

4.5 Діаграма послідовності

Наведемо діаграму в частині графічного матеріалу. Діаграму послідовності було згенеровано у Visual Studio 2013. Для побудови даної діаграми було обрано частину коду, яка реалізує алгоритм знаходження оптимального захисту системи.

4.6 Діаграма розгортання

Наведемо діаграму розгортання в частині графічного матеріалу. Дана діаграма відображає основні обчислювальні вузли програми під час її роботи.

Висновок до розділу

В даному розділі був здійснений опис усіх програмних пакетів, що були використанні при розробці програмного продукту, а також наведено та описано схему архітектури ПЗ, побудовано основні діграми, а саме діаграми класів, послідовності та розгортання.

5 ТЕХНОЛОГІЧНИЙ РОЗДІЛ

Программа відкриває вікно у браузері. Робота користувача починається з головної сторінки. Програма містить вкладки «*Матриця», «Вхідні дані», «Алгоритм», «Контакти».* Користувач повинен послідовно переходити з однієї вкладки навігації на іншну під час роботи з програмою.

Для початку нашого дослідження ми маємо 18 типів об’єктів, 37 типів загроз та 12 механізмів захисту. Для початку за допомогою програми GraphViz будуємо граф, що покаже залежність між об’єктами та загрозами на рисунку 5.1.

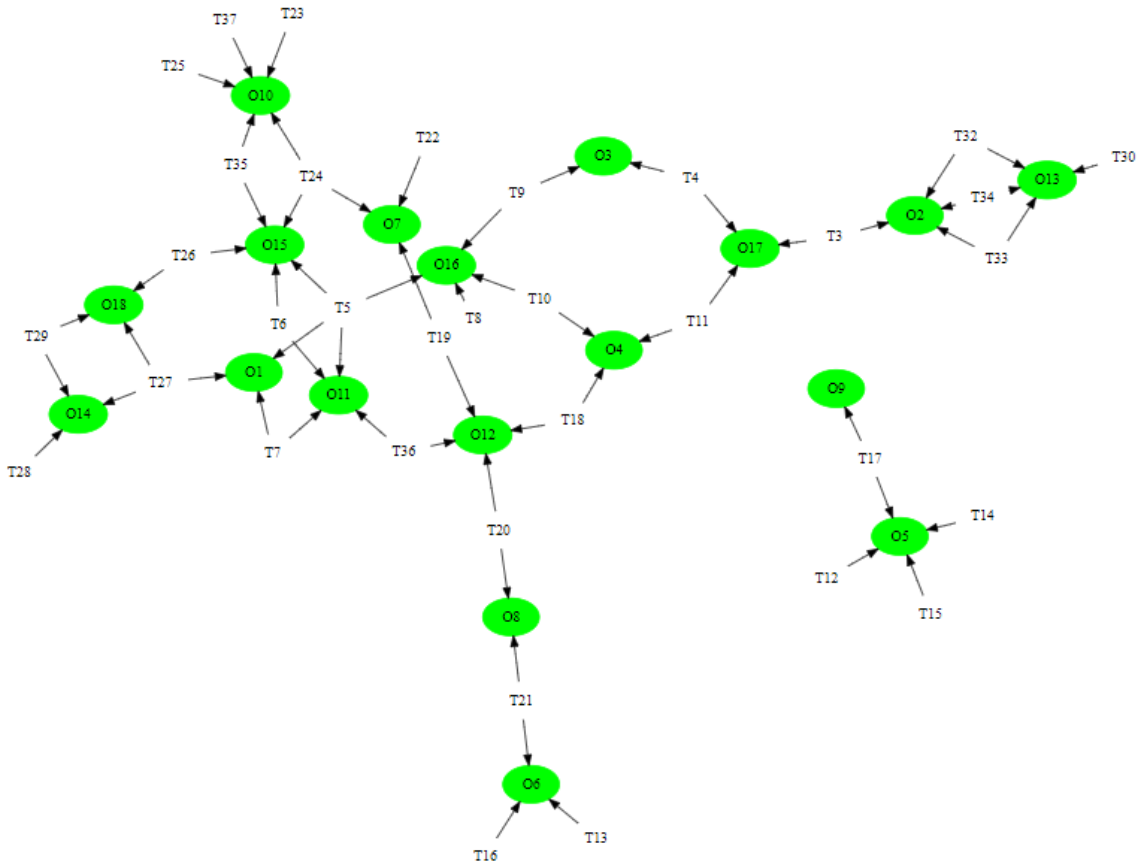


Рисунок 5.1 – Граф залежності об’єктів та загроз

Аналогічно будуємо граф, що показує зв’язок між загрозами та механізмами захисту на рисунку 5.2.

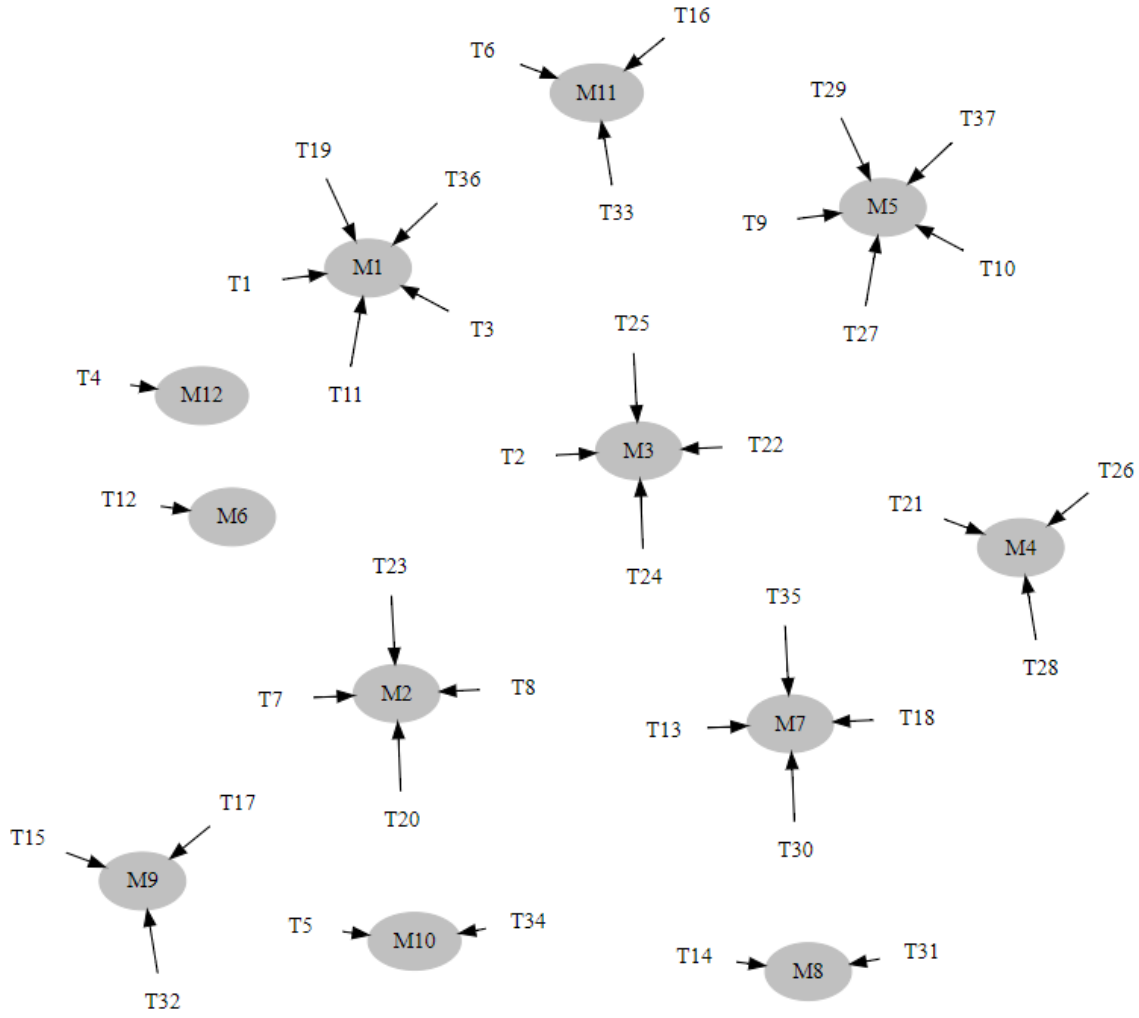
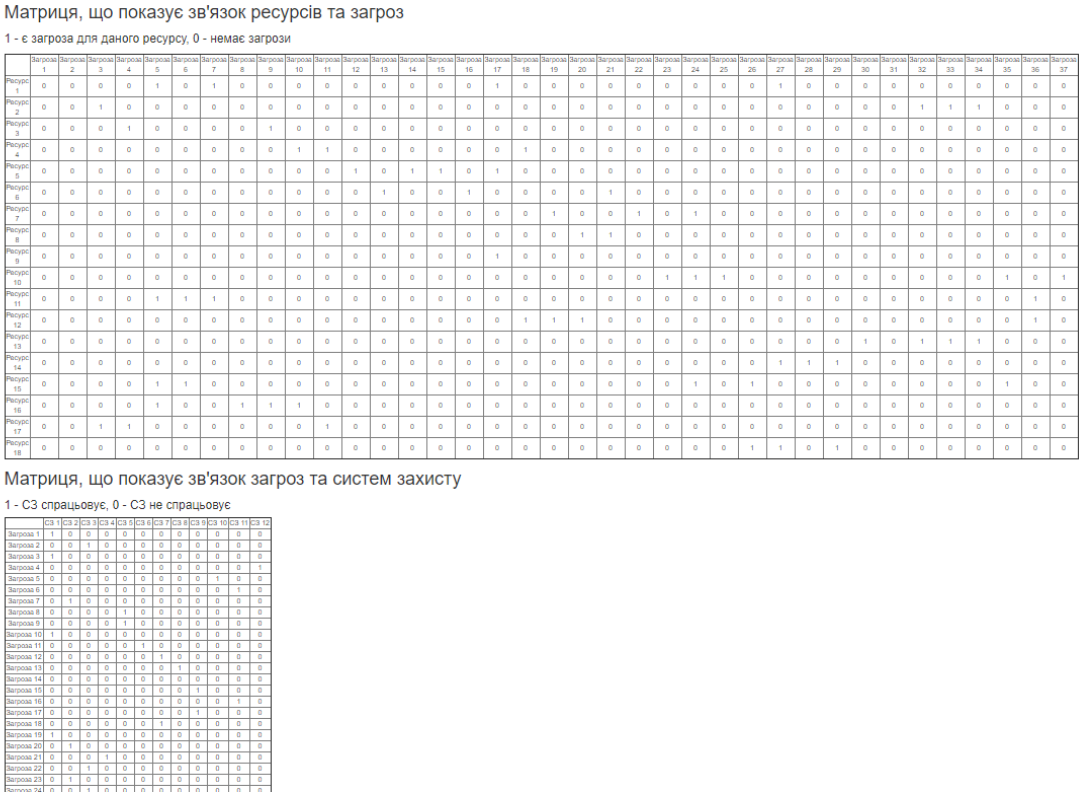


Рисунок 5.2 – Граф залежності загроз та механізмів захисту

Тепер за цими графами будуємо матриці суміжності, використовуючи створений програмний продукт. Перейшовши на вкладку «Матриця» отримали результат на рисунку 5.3.


Матриця, що показує зв'язок ресурсів та загроз

1 - є загроза для даного ресурсу, 0 - немає загрози

	Загроза 1	Загроза 2	Загроза 3	Загроза 4	Загроза 5	Загроза 6	Загроза 7	Загроза 8	Загроза 9	Загроза 10	Загроза 11	Загроза 12	Загроза 13	Загроза 14	Загроза 15	Загроза 16	Загроза 17	Загроза 18	Загроза 19	Загроза 20	Загроза 21	Загроза 22	Загроза 23	Загроза 24	Загроза 25	Загроза 26	Загроза 27	Загроза 28	Загроза 29	Загроза 30	Загроза 31	Загроза 32	Загроза 33	Загроза 34	Загроза 35	Загроза 36	Загроза 37
Ресурс 1	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
Ресурс 2	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0
Ресурс 3	0	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Ресурс 4	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Ресурс 5	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Ресурс 6	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Ресурс 7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
Ресурс 8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Ресурс 9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Ресурс 10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	1	0	1	0
Ресурс 11	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
Ресурс 12	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
Ресурс 13	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	1	0	0	0
Ресурс 14	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0
Ресурс 15	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	1	0	0	0
Ресурс 16	0	0	0	0	1	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Ресурс 17	0	0	1	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Ресурс 18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	1	0	0	0	0	0	0	0	0

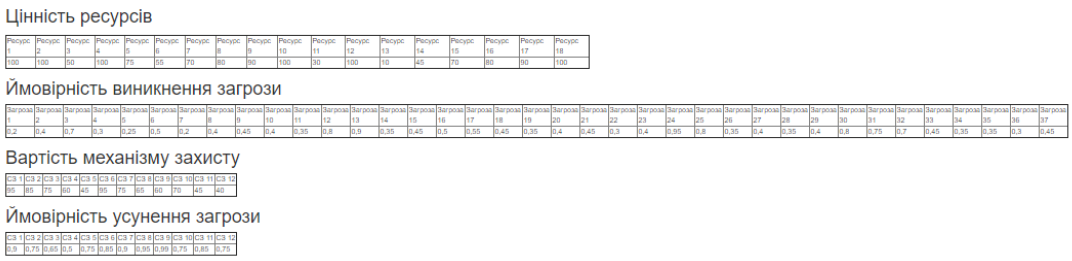
Матриця, що показує зв'язок загроз та систем захисту

1 - СЗ спрацьовує, 0 - СЗ не спрацьовує

	СЗ 1	СЗ 2	СЗ 3	СЗ 4	СЗ 5	СЗ 6	СЗ 7	СЗ 8	СЗ 9	СЗ 10	СЗ 11	СЗ 12
Загроза 1	1	0	0	0	0	0	0	0	0	0	0	0
Загроза 2	0	1	0	0	0	0	0	0	0	0	0	0
Загроза 3	1	0	0	0	0	0	0	0	0	0	0	0
Загроза 4	0	0	0	0	0	0	0	0	0	0	0	1
Загроза 5	0	0	0	0	0	0	0	0	1	0	0	0
Загроза 6	0	0	0	0	1	0	0	0	0	0	1	0
Загроза 7	0	1	0	0	0	0	0	0	0	0	0	0
Загроза 8	0	0	0	0	1	0	0	0	0	0	0	0
Загроза 9	0	0	0	0	1	0	0	0	0	0	0	0
Загроза 10	1	0	0	0	0	0	0	0	0	0	0	0
Загроза 11	0	0	0	0	0	1	0	0	0	0	0	0
Загроза 12	0	0	0	0	0	0	1	0	0	0	0	0
Загроза 13	0	0	0	0	0	0	0	1	0	0	0	0
Загроза 14	0	0	0	0	0	0	0	0	0	0	0	0
Загроза 15	0	0	0	0	0	0	0	0	1	0	0	0
Загроза 16	0	0	0	0	0	0	0	0	0	0	1	0
Загроза 17	0	0	0	0	0	0	0	1	0	0	0	0
Загроза 18	0	0	0	0	0	0	1	0	0	0	0	0
Загроза 19	1	0	0	0	0	0	0	0	0	0	0	0
Загроза 20	0	1	0	0	0	0	0	0	0	0	0	0
Загроза 21	0	0	0	1	0	0	0	0	0	0	0	0
Загроза 22	0	0	1	0	0	0	0	0	0	0	0	0
Загроза 23	0	1	0	0	0	0	0	0	0	0	0	0
Загроза 24	0	0	1	0	0	0	0	0	0	0	0	0

Рисунок 5.3 – Утворені матриці суміжності

Переходимо до роботи з вхідними даними, які записані у текстовому файлі data.txt, а саме ймовірність виникнення загрози та ймовірність її усунення, а також цінність ресурсів та вартість встановлення кожного механізму захисту. Виводимо ці дані на екран на рисунку 5.4.


Цінність ресурсів

Ресурс 1	Ресурс 2	Ресурс 3	Ресурс 4	Ресурс 5	Ресурс 6	Ресурс 7	Ресурс 8	Ресурс 9	Ресурс 10	Ресурс 11	Ресурс 12	Ресурс 13	Ресурс 14	Ресурс 15	Ресурс 16	Ресурс 17	Ресурс 18
100	100	50	100	75	55	70	80	90	100	30	100	10	45	70	80	90	100

Ймовірність виникнення загрози

Загроза 1	Загроза 2	Загроза 3	Загроза 4	Загроза 5	Загроза 6	Загроза 7	Загроза 8	Загроза 9	Загроза 10	Загроза 11	Загроза 12	Загроза 13	Загроза 14	Загроза 15	Загроза 16	Загроза 17	Загроза 18	Загроза 19	Загроза 20	Загроза 21	Загроза 22	Загроза 23	Загроза 24	Загроза 25	Загроза 26	Загроза 27	Загроза 28	Загроза 29	Загроза 30	Загроза 31	Загроза 32	Загроза 33	Загроза 34	Загроза 35	Загроза 36	Загроза 37
0.2	0.4	0.7	0.3	0.25	0.5	0.2	0.4	0.45	0.4	0.35	0.8	0.9	0.35	0.45	0.5	0.55	0.45	0.35	0.4	0.45	0.3	0.4	0.95	0.8	0.35	0.4	0.35	0.4	0.8	0.75	0.7	0.45	0.35	0.35	0.3	0.45

Вартість механізму захисту

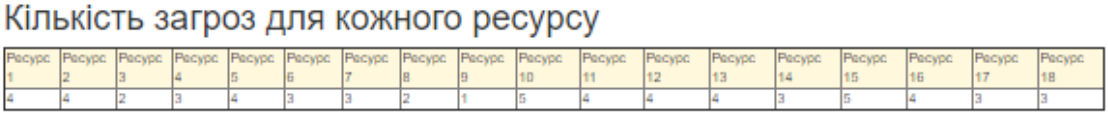
СЗ 1	СЗ 2	СЗ 3	СЗ 4	СЗ 5	СЗ 6	СЗ 7	СЗ 8	СЗ 9	СЗ 10	СЗ 11	СЗ 12
95	85	75	60	45	95	75	65	80	70	45	40

Ймовірність усунення загрози

СЗ 1	СЗ 2	СЗ 3	СЗ 4	СЗ 5	СЗ 6	СЗ 7	СЗ 8	СЗ 9	СЗ 10	СЗ 11	СЗ 12
0.9	0.75	0.65	0.5	0.75	0.85	0.9	0.95	0.99	0.75	0.85	0.75

Рисунок 5.4 – Вхідні дані

Натиснувши вкладку меню «Алгоритм» переходимо до розрахунків. Спочатку розраховуємо кількість загроз для кожного об’єкту на рисунку 5.5.


Кількість загроз для кожного ресурсу

Ресурс 1	Ресурс 2	Ресурс 3	Ресурс 4	Ресурс 5	Ресурс 6	Ресурс 7	Ресурс 8	Ресурс 9	Ресурс 10	Ресурс 11	Ресурс 12	Ресурс 13	Ресурс 14	Ресурс 15	Ресурс 16	Ресурс 17	Ресурс 18
4	4	2	3	4	3	3	2	1	5	4	4	4	3	5	4	3	3

Рисунок 5.5 – Розрахунок кількості загроз для кожного об’єкту

Далі починає працювати розроблений алгоритм і програма виділяє підграфи. За нашими даними ми отримали 8 підграфів. Ділення на підграфи продовжується до тих пір, поки не будуть використані усі наявні об’єкти на рисунку 5.6.

Утворені підграфи

Підграф 1	15	23	24	42	44	53	1	11	16	7	10	18	25	35	45	54	26	27	28	37	40	41	43	55	47				
Підграф 2	10	41	42	43	53	55	7	15	37	40	23	24	44																
Підграф 3	16	23	26	27	28	1	11	15	3	4	25	35	45	24	54	42	44	53	22	29	36								
Підграф 4	13	48	50	51	52	2	21																						
Підграф 5	12	36	37	38	54	4	7	8	11	28	29	40	42	39	23	24	25												
Підграф 6	11	23	24	25	54	1	15	16	12	35	45	42	44	53	26	27	28	36	37	38									
Підграф 7	5	30	32	33	35	1	9	23	25	45																			
Підграф 8	2	21	50	51	52	17	13	22	29	48																			

Рисунок 5.6 – Виділення підграфів за допомогою створеного програмного забезпечення

Утворені підграфи є дводольними, тобто поки що ми розглядаємо зв’язок тільки об’єктів та загроз. Для наочності зобразимо утворені підграфи за допомогою графічного редактора GraphViz на рисунках 5.7 – 5.13.

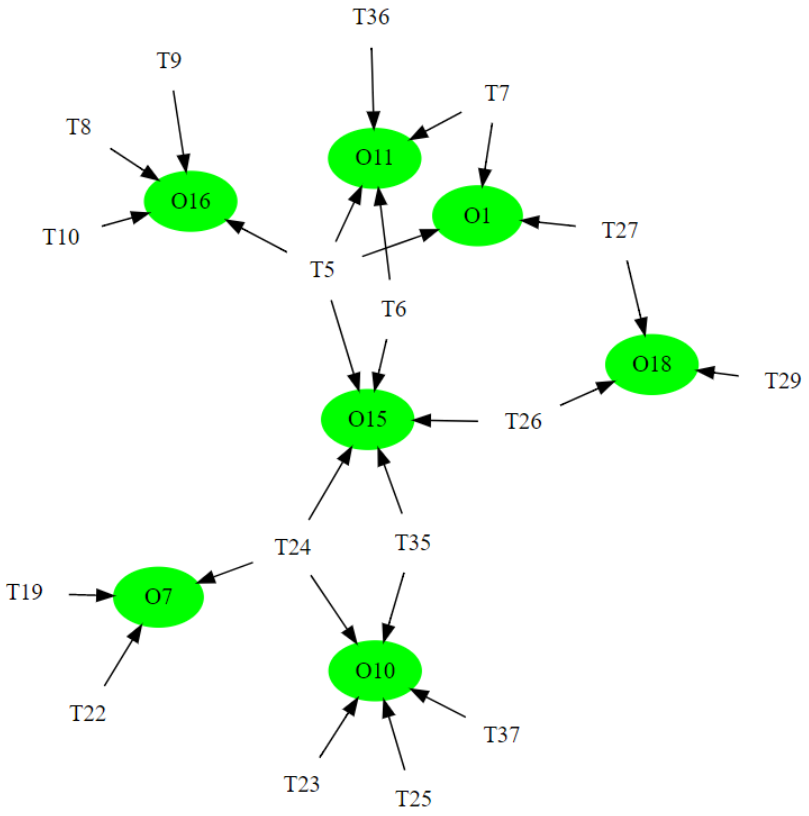


Рисунок 5.7 – Дводольний підграф 1

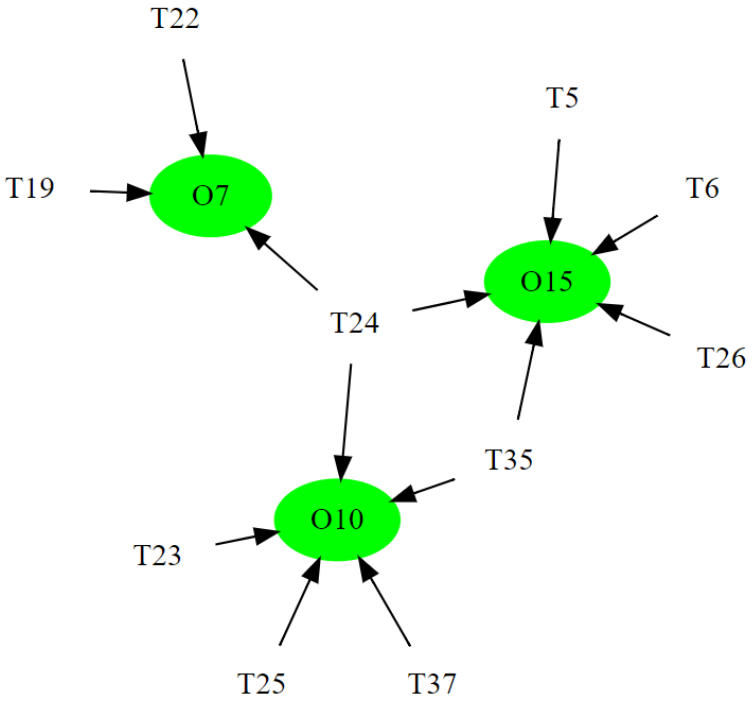


Рисунок 5.8 – Дводольний підграф 2

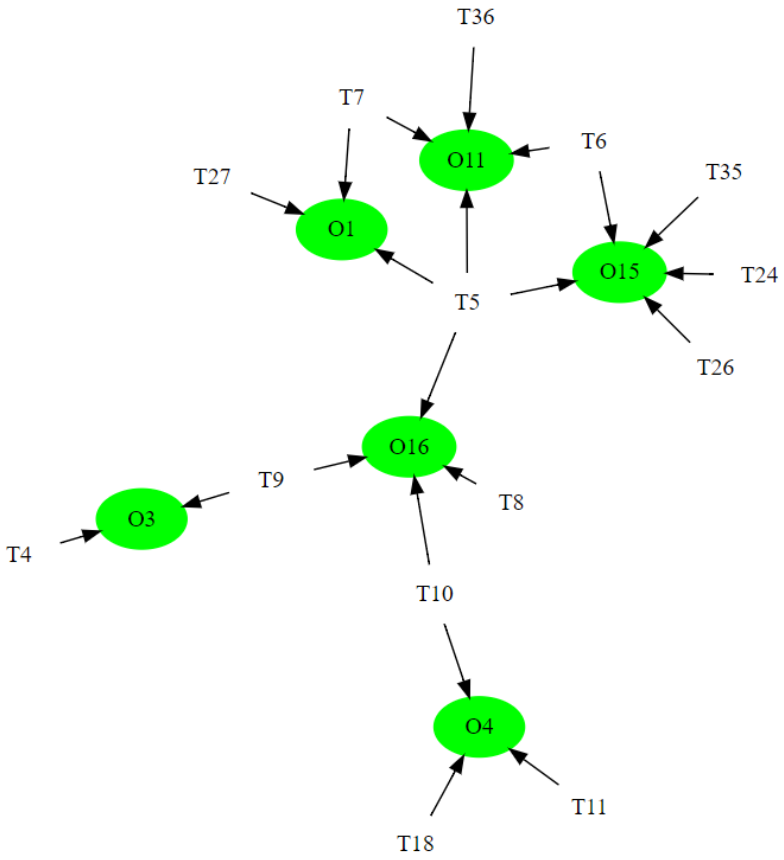


Рисунок 5.9 – Дводольний підграф 3

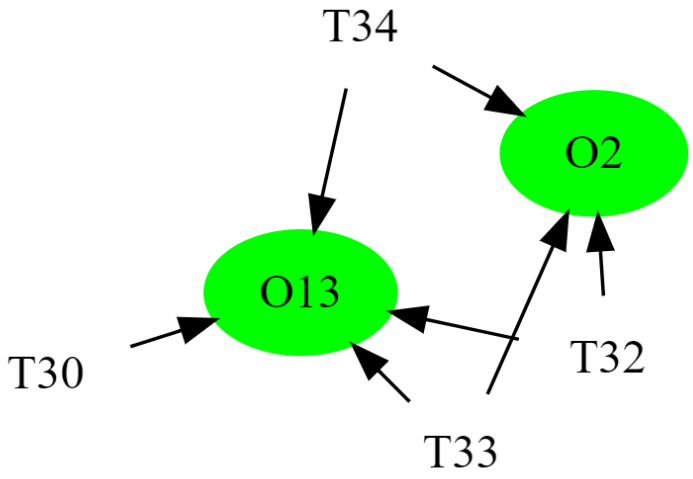


Рисунок 5.10 – Дводольний підграф 4

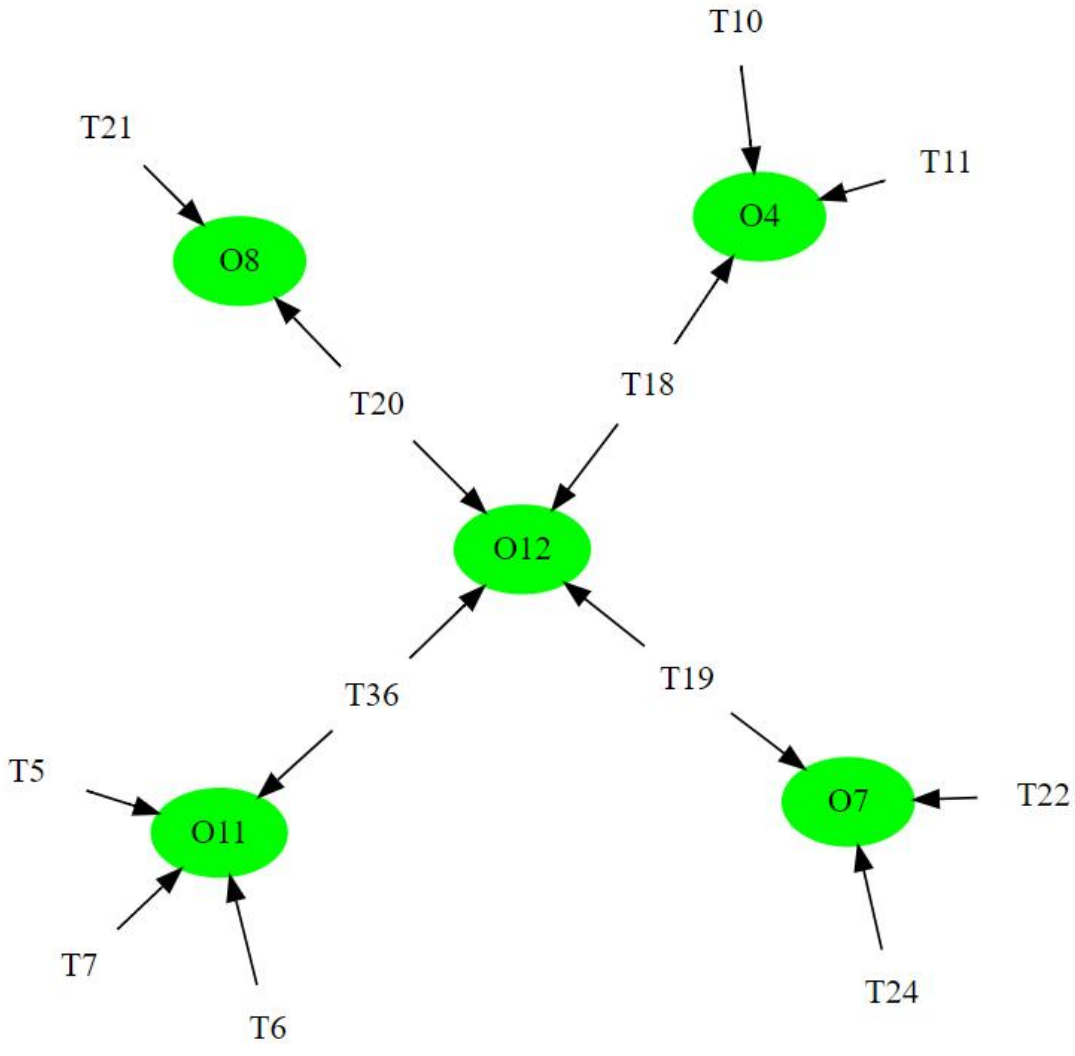


Рисунок 5.11 – Дводольний підграф 5

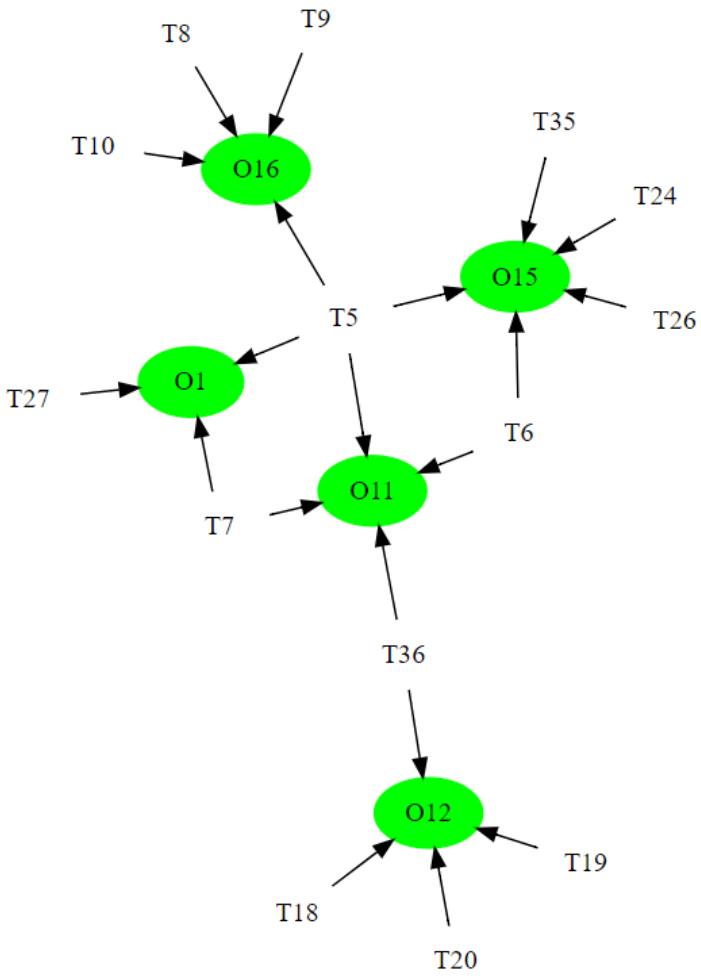


Рисунок 5.11 – Дводольний підграф 6

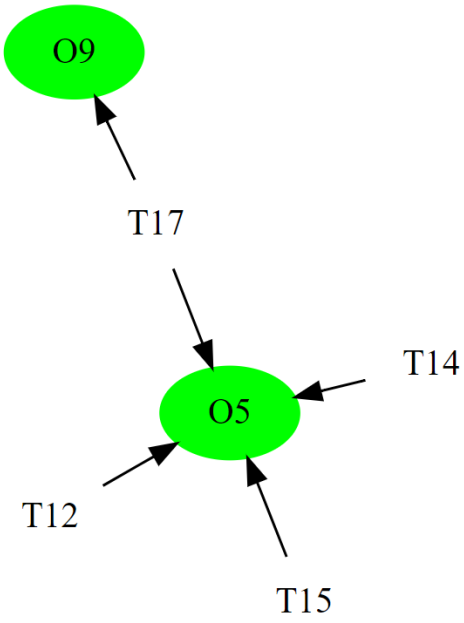


Рисунок 5.12 – Дводольний підграф 7

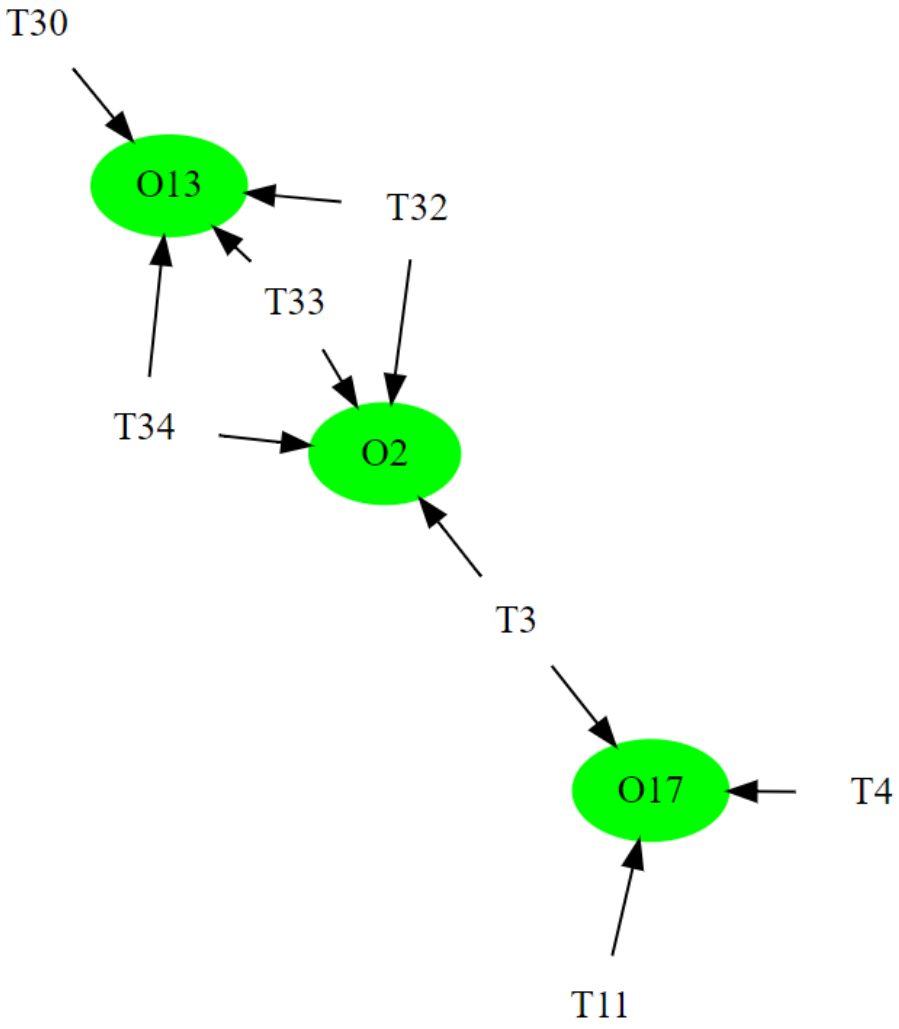


Рисунок 5.13 – Дводольний підграф 8

Далі проводимо оцінку втрат при настанні загроз для кожного типу ресурсів та дивимось наскільки зменшяться втрати при встановленні захисту на рисунку 5.14.

Втрати для кожного ресурсу без МЗ

Ресурс 1	Ресурс 2	Ресурс 3	Ресурс 4	Ресурс 5	Ресурс 6	Ресурс 7	Ресурс 8	Ресурс 9	Ресурс 10	Ресурс 11	Ресурс 12	Ресурс 13	Ресурс 14	Ресурс 15	Ресурс 16	Ресурс 17	Ресурс 18
55	70	22,5	45	60	49,5	66,5	36	49,5	95	15	45	8	18	66,5	36	63	40

Втрати для кожного ресурсу при застосуванні МЗ

Ресурс 1	Ресурс 2	Ресурс 3	Ресурс 4	Ресурс 5	Ресурс 6	Ресурс 7	Ресурс 8	Ресурс 9	Ресурс 10	Ресурс 11	Ресурс 12	Ресурс 13	Ресурс 14	Ресурс 15	Ресурс 16	Ресурс 17	Ресурс 18
13,75	17,5	5,625	11,25	15	12,375	16,625	9	12,375	23,75	3,75	11,25	2	4,5	16,625	9	15,75	10

Рисунок 5.14 – Оцінка втрат при здійсненні загроз

Тепер доповнимо наші підграфи механізмами захисту та отримаємо тридольні підграфи. Аналогічно зобразимо їх за допомогою програми GraphViz на рисунках 5.15 – 5.22.

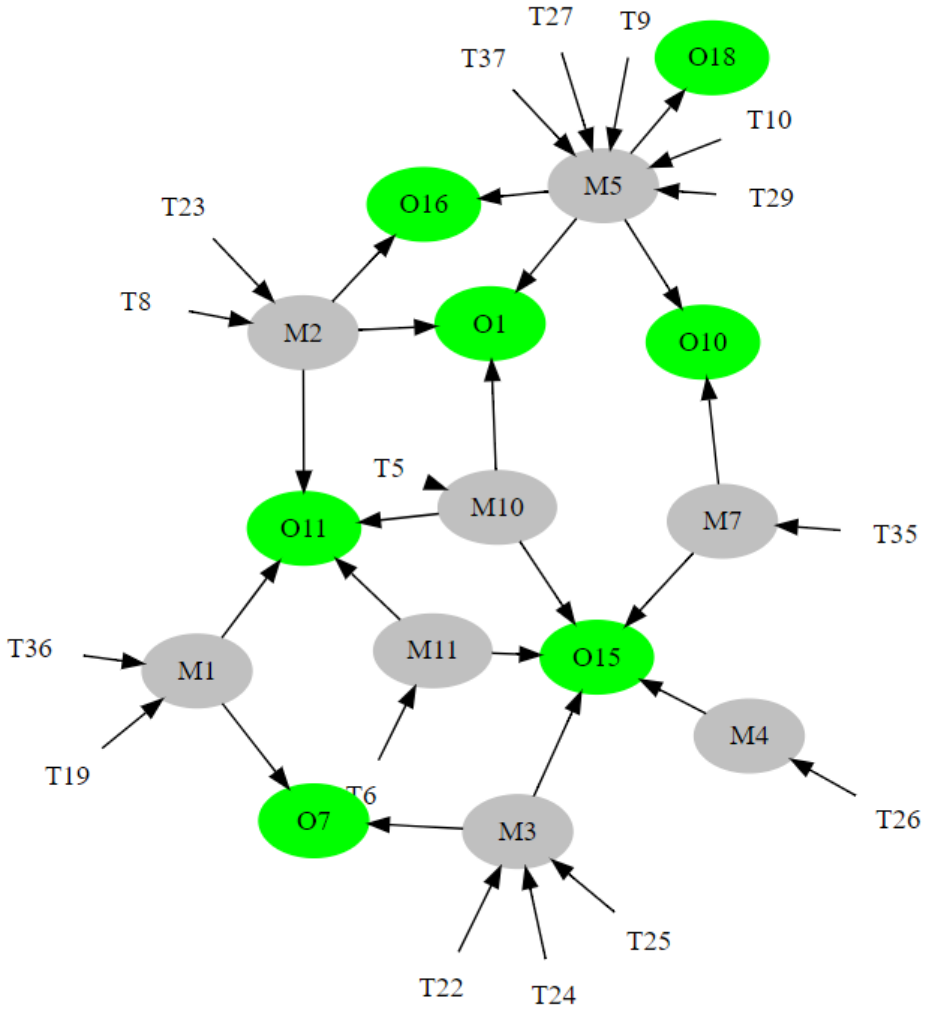


Рисунок 5.15 – Тридольний підграф 1

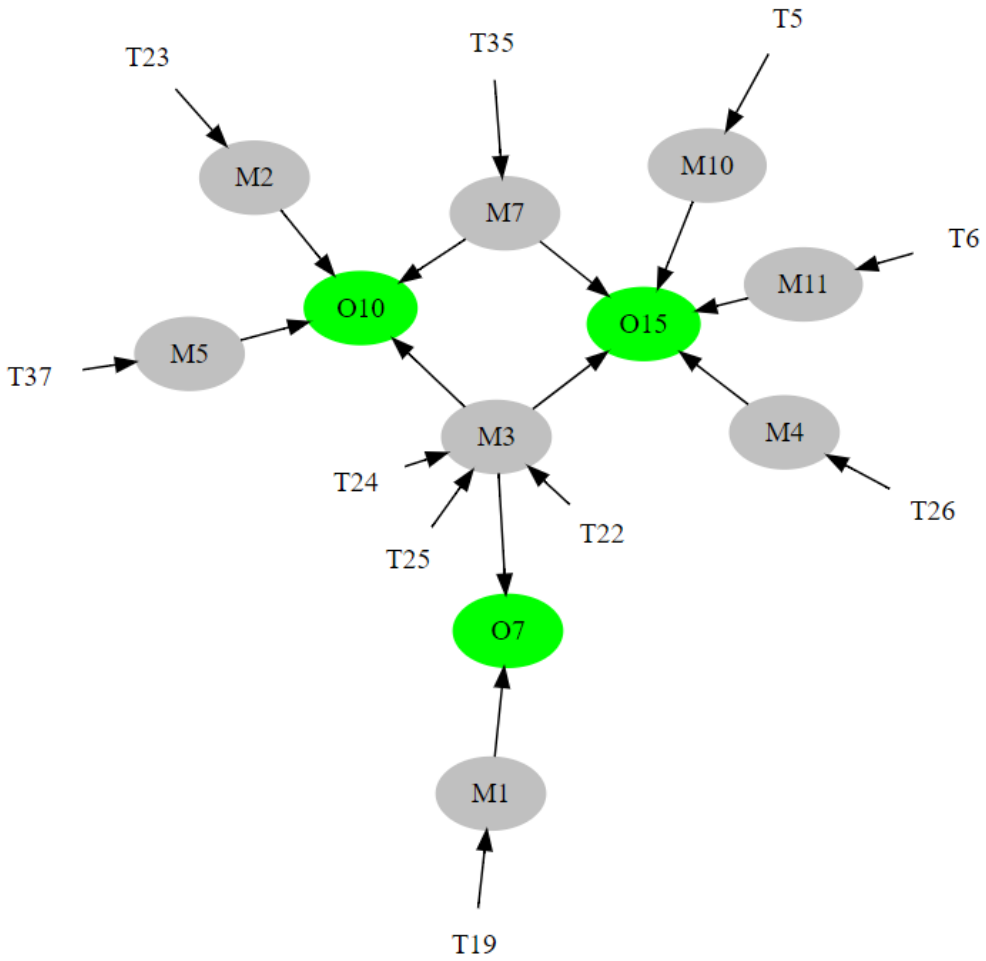


Рисунок 5.16 – Тридольний підграф 2

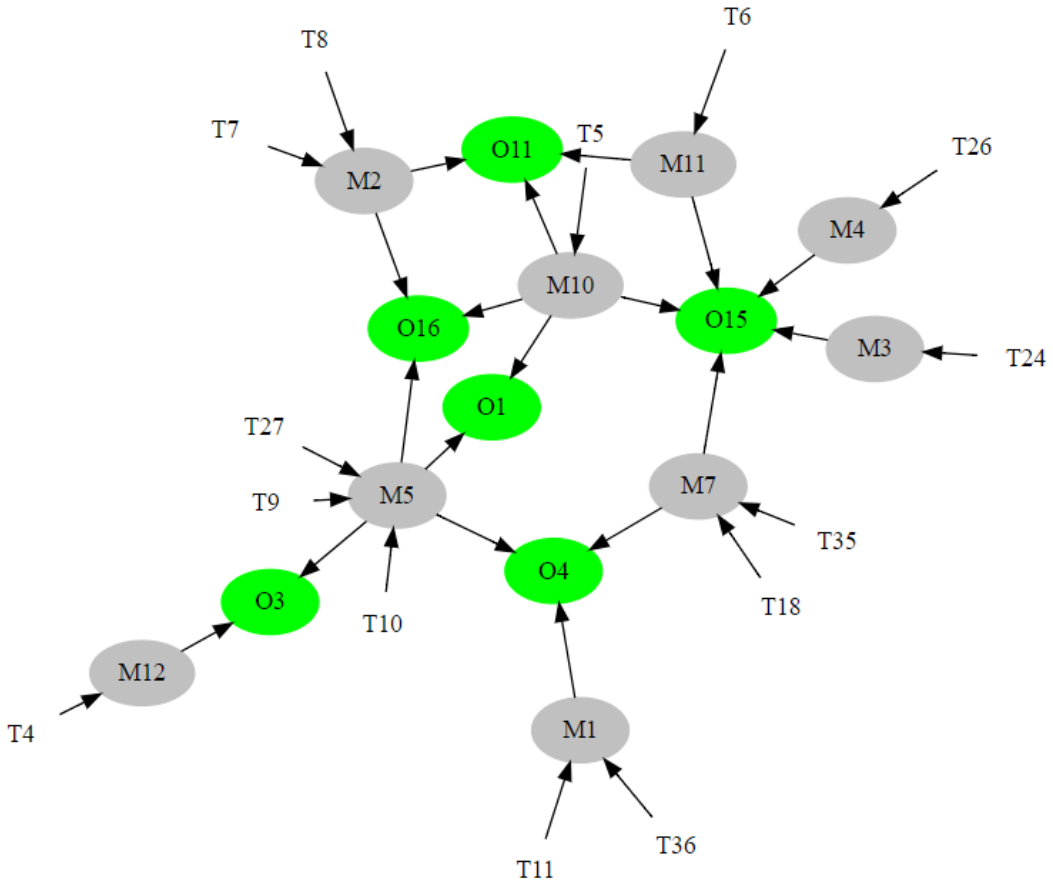


Рисунок 5.17 – Тридольний підграф 3

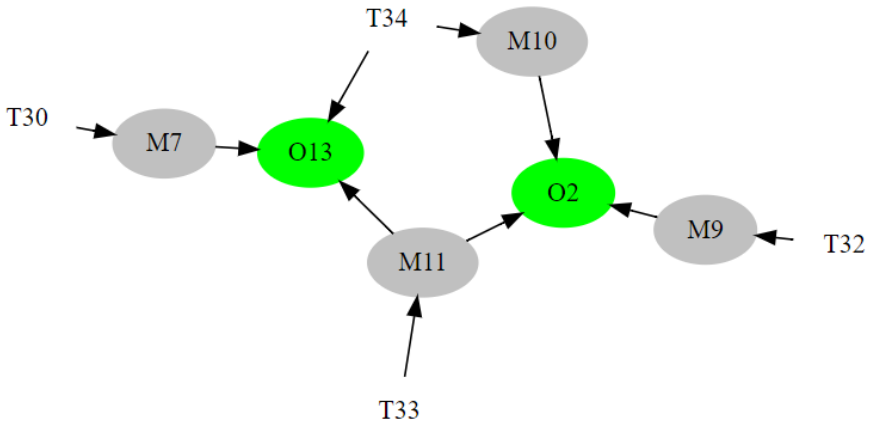


Рисунок 5.18 – Тридольний підграф 4

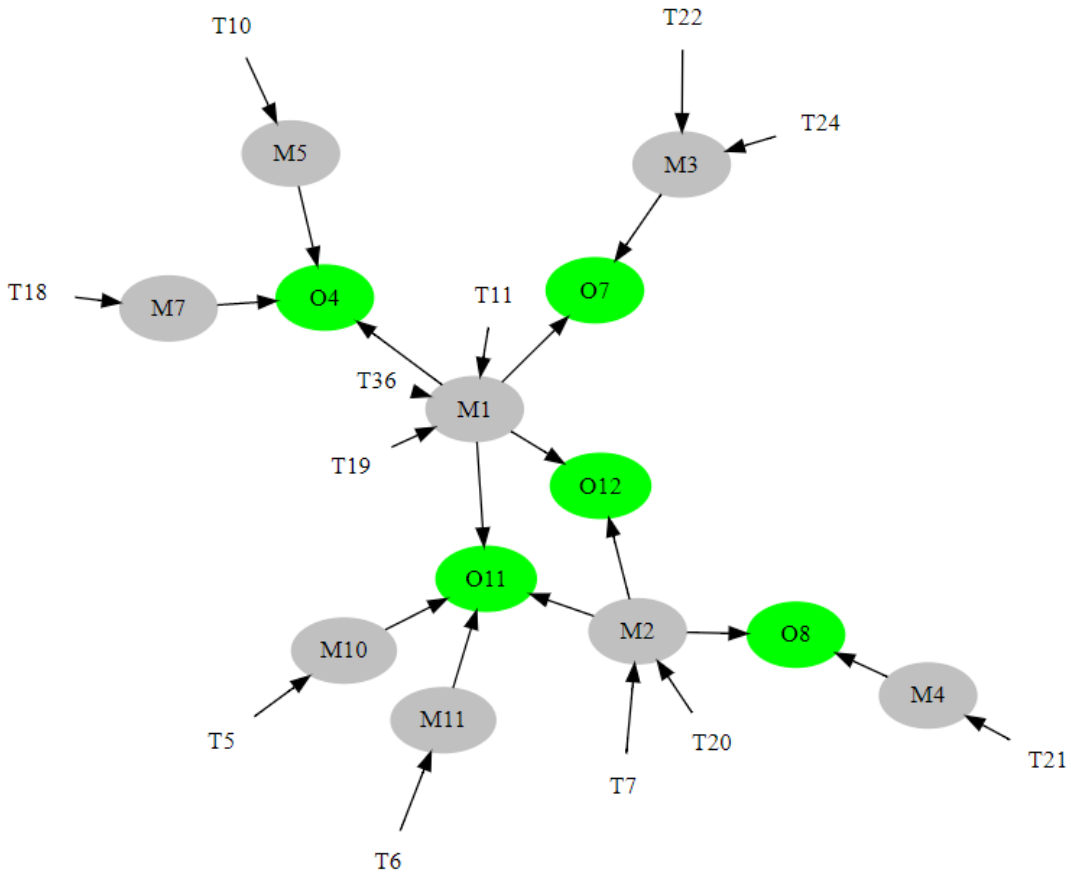


Рисунок 5.19 – Тридольний підграф 5

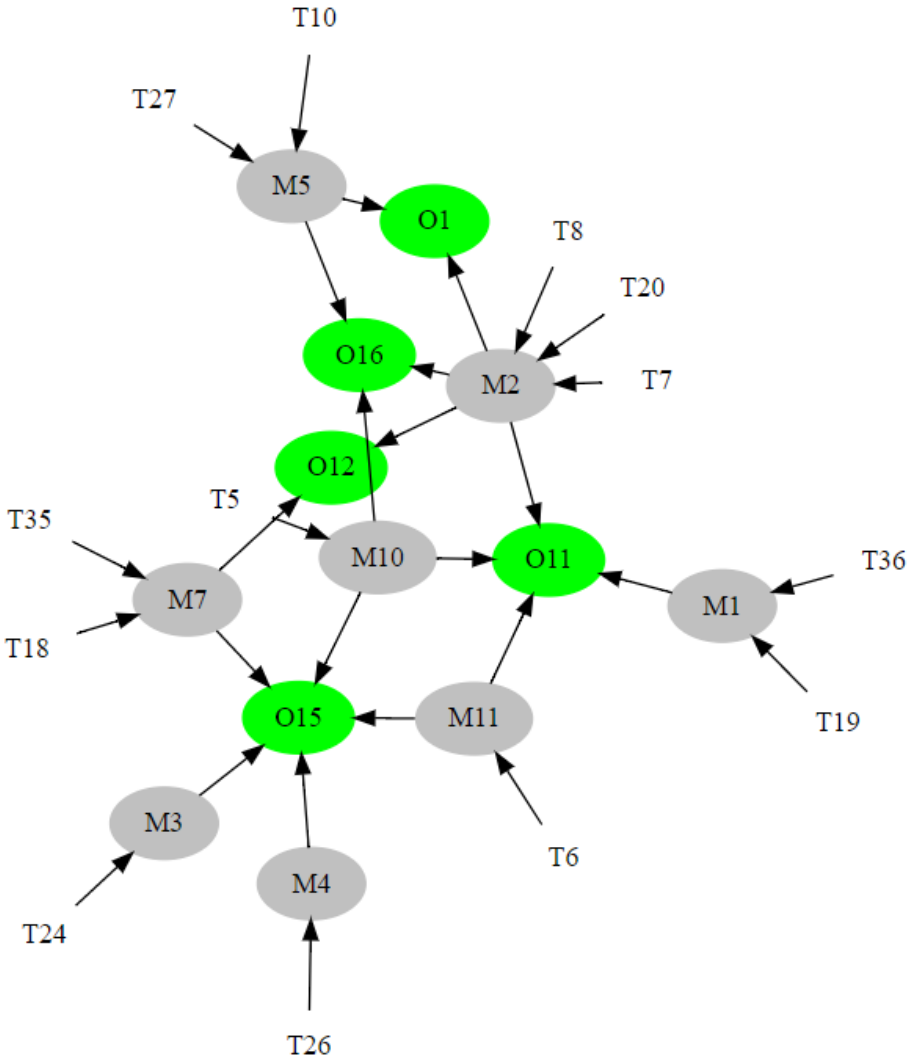


Рисунок 5.20– Тридольний підграф 6

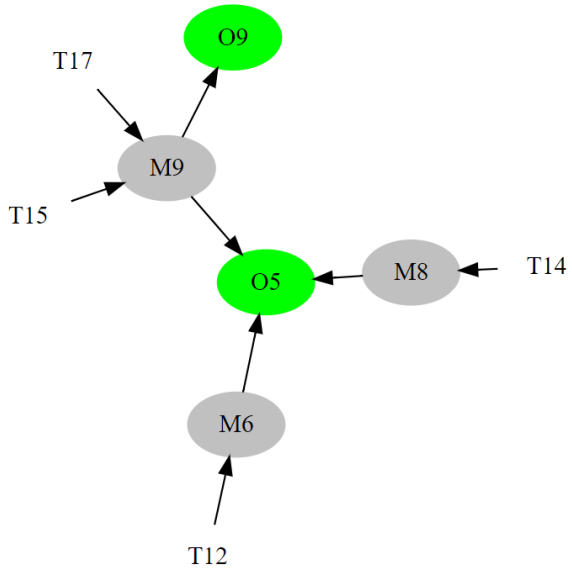


Рисунок 5.21 – Тридольний підграф 7

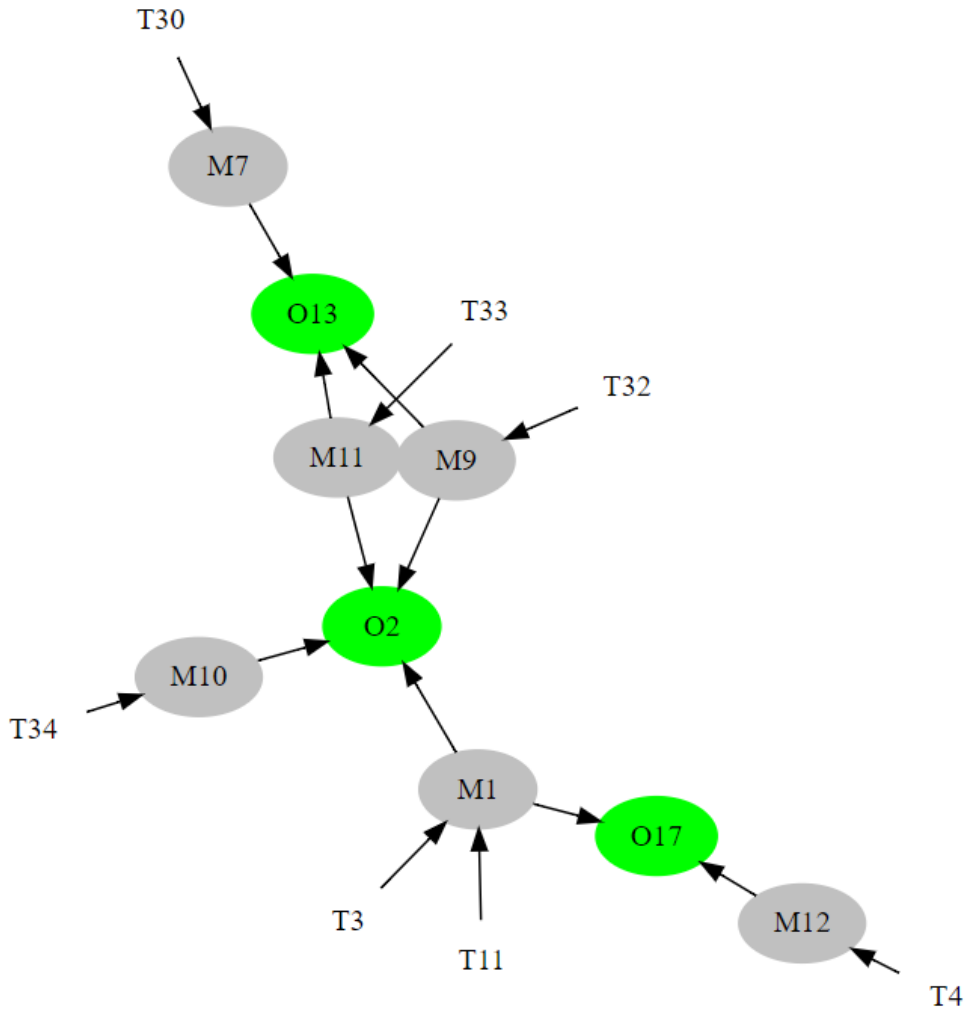


Рисунок 5.22 – Тридольний підграф 8

Далі проводимо оцінку механізмів захисту, а також розраховуємо оцінку остаточної вартості встановлення захисту для кожного типу об’єкта на рисунку 5.23.

Кількість загроз, на які впливає МЗ

МЗ 1	МЗ 2	МЗ 3	МЗ 4	МЗ 5	МЗ 6	МЗ 7	МЗ 8	МЗ 9	МЗ 10	МЗ 11	МЗ 12
5	3	4	3	5	1	4	2	3	2	3	1

Оцінка МЗ

МЗ 1	МЗ 2	МЗ 3	МЗ 4	МЗ 5	МЗ 6	МЗ 7	МЗ 8	МЗ 9	МЗ 10	МЗ 11	МЗ 12
19	28,33	18,75	20	9	95	18,75	32,5	20	35	15	40

Вартість встановлення МЗ для кожного об'єкту

Ресурс1	Ресурс2	Ресурс3	Ресурс4	Ресурс5	Ресурс6	Ресурс7	Ресурс8	Ресурс9	Ресурс10	Ресурс11	Ресурс12	Ресурс13	Ресурс14	Ресурс15	Ресурс16	Ресурс17	Ресурс18
92,33	89	49	132,75	58,75	67,5	56,5	48,33	20	93,58	97,33	85,08	88,75	38	107,5	72	154	38

Рисунок 5.23 – Оцінки МЗ

Тепер порівнявши втрати для кожного типу об’єкту та вартість встановлення для нього захисту можемо стверджувати про доцільність його встановлення.

Висновок до розділу

В цьому розділі наведено керівництво користувача та механізм роботи програмного продукту.

ЗАГАЛЬНІ ВИСНОВКИ

У ході виконання дипломного проекту були детально розглянуті питання, які виникають в процесі визначення оптимального механізму захисту для системи в цілому.

Був проведений повний аналіз предметного середовища, визначено основні цілі і задачі розробки.

На основі даних, отриманих в процесі аналізу, було сформульовано математичну задачу. Аналіз усіх наявних видів ресурсів, загроз та механізмів захистів, виділення підграфів допомагає визначити кількісні оцінки та зробити висновки про доцільність встановлення того чи іншого механізму захисту, а також дозволяє запобігти втратам, які можуть виникнути внаслідок здійснення загроз.

Для розробки програмного забезпечення була використана мова C# програмної платформи Microsoft Visual Studio 2013, а також в якості додаткового програмного забезпечення – GraphViz для побудови графів та підграфів для більш наочного сприйняття інформації.

Був виконаний детальний опис роботи програми та керівництво користувача, а також здійснено випробування програмного продукту, що показує аналіз системи на прикладі наявного переліку типів ресурсів, загроз, механізмів захисту. В програмі було розраховано кількісні оцінки втрат ресурсів при здійсненні загрози, втрати при застосуванні механізму захисту, а також повна вартість встановлення механізму захисту для кожного типу ресурсу. А також за допомогою побудованих підграфів ми зможемо більш наочно побачити та проаналізувати рівень безпеки ресурсів інформаційної системи.

ПЕРЕЛІК ПОСИЛАНЬ

1. Нестеренко О.В. Безпека інформаційного простору державної влади. Технологічні основи. –К.: Наукова думка, 2009. – 352с.
2. Качинський А. Б. Безпека, загрози і ризик: наукові концепції та математичні методи / А.Б. Качинський // Інститут проблем національної безпеки; Національна академія Служби безпеки України. — К., 2004. — 472с.
3. Нетесін І.Є. Модели безопасности и зашиты в распределенных компьютерных средах / І.Є. Нетесін // Проблемы программирования, 2000, № 3-4. – С. 148-158.
4. Юдін О.К. Державні інформаційні ресурси. Методологія побудови класифікатора загроз: монографія / О.К. Юдін,С. С. Бучик. — К.: НАУ, 2015. — 214 с.
5. Стрижак О.Є. Засоби онтологічної інтеграції і супроводу розподілених просторових та семантичних інформаційних ресурсів / О.Є. Стрижак // Екологічна безпека та природокористування. Збірник наукових праць. Інститут телекомунікацій і глобального інформаційного простору НАН України. – 2013,вип. 12. – С. 166–177.
6. Качинський А.Б. Ієрархія факторів типових сценаріїв реалізації DDos-атак/ А.Б. Качинський, В.М. Ткач, А.А. Поденко // Математичне моделювання в економіці, 2017. – №1-2. – С. 17-30, 2018. – №1. – С. 31-48.
7. Хнигічева А.М. Моделювання безпеки складних інформаційно-комунікаційних систем із використанням логіко-ймовірнісного методу / А.М.Хнигічева, О.М. Новіков, А.О. Тимошенко // Наукові вісті Національного технічного університету України Київський політехнічний інститут, 2010. –Вип.6. – С. 70-77.

8. Пустовіт О.С. Про застосування алгебраїчної комбінаторики до проблем кодування та криптографії / О.С. Пустовіт, В.О. Устименко // Математичне моделювання в економіці, №1-2, 2017. - С. 31-46.
9. Палагин А.В. К вопросу системно-онтологической интеграции знаний предметной области / А.В. Палагин, Н.Г. Петренко //Математические машины и системы, 2007. – № 3,4. – С. 63–75.
10. Хоффман Л.Дж. Современные методы защиты информации / Л.Дж. Хоффман. Пер. с англ. / М.: Советское радио, 1980. – 264 с.
11. Кормен Т. Алгоритмы. Построение и анализ / Т. Кормен, Ч. Лейзерсон, Р. Ривест, К. Штайн / пер. с англ. – 2-е изд. – Москва–Санкт-Петербург–Киев: Издательский дом «Вильямс», 2013. – 1296 с.
12. Рейнгольд Э. Комбинаторные алгоритмы. Теория и практика / Э. Рейнгольд, Ю. Нивергельт, Н. Део / пер. с англ. – М: «Мир», 1980. – 478 с.
13. Харари Ф. Теория графов / Ф. Харари / пер. с англ. – М: «Мир», 1973. – 302 с.
14. Берзтисс А.Т. Структуры данных / А.Т. Берзтисс / пер. с англ. – М: «Статистика», 1974. – 408 с.

Додаток А

Тексти програмного коду

Графова модель безпеки ресурсів інформаційної системи

(Найменування програми (документа))

DVD-R

(Вид носія даних)

15 арк, 88,5 Мб

(Обсяг програми (документа) , арк.,) Кб)

Київ – 2019 року

Algoritm.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.IO;


namespace Course_Work.Models
{
    public class Algoritm
    {
        public static int amount_ver = 0;
        public static int amount_dependency = 0;
        public static List<List<int>> incendent = new List<List<int>>();
        public static List<List<double>> ves = new List<List<double>>();
        public static List<double> ver = new List<double>();
        public static List<double> row = new List<double>();
        public static List<int> rowi = new List<int>();
        public static double[] lost= new double[18];
        public static double[] lost_safe = new double[18];
        public static double[] result_cost = new double[18];
       public static  int k = 0;
       public static double[] important = new double[19];
       public static double[] problem = new double[38];
       public static double[] safe = new double[13];
       public static double[] cost = new double[13];
       public static int[] treats_count = new int[20];
       public static int[] treats_count_safe = new int[20];
       public static double[] mark_safe = new double[20];

        public int Variables { get; set; }

        public Course_Work.Controllers.AlgoritmController AlgoritmController
        {
            get
            {
                throw new System.NotImplementedException();
            }
            set
            {
            }
        }

        public static List<List<int>> Podgraf = new List<List<int>>();
        public static List<int> podgraf1 = new List<int>();
        public static void Incedent(int a, int b)
        {
            for (int i = 0; i < amount_ver; i++)
                for (int j = 0; j < amount_ver; j++)
                {
                    if (i == (a - 1) && j == b - 1)
                    {
                        incendent[i][j] = 1;

                        incendent[j][i] = 1;
                    }

                }
        }
        public static void ReadFile()
        {
```

```
            using (var stream =
File.OpenText("C:/Users/User/Desktop/Course_Work/Course_Work/App_start/App_Data/graf.txt"
))
            {
                string line;
                int first = 0;
                while ((line = stream.ReadLine()) != null)
                {
                    var columns = line.Split(' ');

                    if (first == 0)
                    {
                        amount_ver = Int32.Parse(columns[0]);
                        first++;
                        amount_dependency = Int32.Parse(columns[1]);
                        for (int i = 0; i < amount_ver; i++)
                        {
                            rowi = new List<int>();
                            for (int j = 0; j < amount_ver; j++) rowi.Add(0);
                            incendent.Add(rowi);

                        }


                    }
                    else
                    {


                        Incedent(Int32.Parse(columns[0]), Int32.Parse(columns[1]));

                    }

                }



            }
        }
        public static void ReadDataFile()
        {
            using (var stream =
File.OpenText("C:/Users/User/Desktop/Course_Work/Course_Work/App_start/App_Data/data.txt"
))
            {
                string line;
                while ((line = stream.ReadLine()) != null)
                {
                    var columns = line.Split(' ');
                    Data(Int32.Parse(columns[0]), Double.Parse(columns[1]));

                }

            }
        }
        public static void Data(int a, double b)
        {
            if (a <= 18)
            {
                important[a - 1] = b;
            }
            if (a > 18 && a <= 55)
```

```
            {
                problem[a - 19] = b;
            }
            if (a > 55)
            {
                safe[a - 56] = b;
            }

        }
        public static void zero_treat()
        {
            for (int i = 0; i < 18; i++)
            {
                treats_count[i] = 0;
            }
        }

        public static void Treat_Count()
        {
            for (int i = 0; i < 18; i++)
            {
                for (int j = 18; j < 55; j++)
                {
                    if (incendent[i][j] == 1) treats_count[i]++;
                }
            }
        }
        public static void PodgrafFound()
        {

            for (int k = 0; k < 8; k++)
            {
                int maximum_treat = treats_count[0];
                int number_res = 0;
                for (int i = 0; i < 18; i++)
                {

                    if (treats_count[i] >= maximum_treat)
                    {
                        maximum_treat = treats_count[i];
                        number_res = i + 1;
                    }
                }
                treats_count[number_res - 1] = 0;
                List<int> podgraf1 = new List<int>();
                podgraf1.Add(number_res);
                for (int j = 18; j < 55; j++)
                {
                    if (incendent[number_res-1][j] == 1) podgraf1.Add(j + 1);
                }
                int count = podgraf1.Count;
                for (int z = 1; z < count; z++)
                {
                    for (int m = 0; m < 18; m++)
                    {   bool flag=true;
                    for (int b = 0; b < podgraf1.Count(); b++ )
                    {
                        if ((m + 1) == podgraf1[b]) flag = false;
                    }
                        if (incendent[m][podgraf1[z]-1] == 1 && flag) podgraf1.Add(m +
1);
```

```
                    }
                }
                for (int y=count; y<podgraf1.Count(); y++)
                {
                    for (int q = 18; q< 55; q++)
                    {
                        bool flag1 = true;
                        for (int w = 0; w < podgraf1.Count(); w++)
                        {
                            if ((q + 1) == podgraf1[w]) flag1 = false;
                        }
                        if (incendent[podgraf1[y] - 1][q] == 1 && flag1) podgraf1.Add(q +
1);
                    }
                }
                    Podgraf.Add(podgraf1);

            }
        }
        public static void Cost(int a, double b)
        {
            cost[a - 56] = b;
        }
        public static void ReadCostFile()
        {
            using (var stream =
File.OpenText("C:/Users/User/Desktop/Course_Work/Course_Work/App_start/App_Data/cost.txt"
))
            {
                string line;
                while ((line = stream.ReadLine()) != null)
                {
                    var columns = line.Split(' ');
                    Cost(Int32.Parse(columns[0]), Double.Parse(columns[1]));

                }
            }
        }

        public static void Treat_Count_Safe()
        {
            for (int i = 18; i < 55; i++)
            {
                for (int j = 55; j < 67; j++)
                {
                    if (incendent[i][j] == 1) treats_count_safe[j-55]++;
                }
            }
        }
        public static void Mark_Safe()
        {
            for (int i = 0; i < 12; i++)
            {
                mark_safe[i] = Math.Round(cost[i] / treats_count_safe[i],2);
            }
        }
        public static void losted()
        {
            for (int i = 0; i < 18; i++)
            {
                double max_lost = 0;
                List<double> losts = new List<double>();
```

```
                for(int j=18; j<55; j++)
            {

                if (incendent[i][j] == 1)
                {
                    losts.Add(important[i] * problem[j - 18]);
                }
                }
                 max_lost = losts[0];
                 for (int k = 0; k < losts.Count(); k++)
                 {
                     if (losts[k] > max_lost)
                     {
                         max_lost = losts[k];
                     }
                 }
            lost[i] = max_lost;
                }
            }
        public static void losted_safe()
        {
            for (int i = 0; i < 18; i++)
            {
                for (int j = 18; j < 55; j++)

                {
                    for (int m = 55; m < 67; m++)
                    {
                        if (incendent[j][m] == 1)
                        {
                            lost_safe[i] = lost[i] * (1 - safe[m - 55]);
                        }
                    }
                }
            }
        }
        public static void Result_cost()
        {
            for (int i = 0; i < 18; i++)
            {
                for (int j = 18; j < 55; j++)
                {
                    if (incendent[i][j] == 1)
                    {
                        for (int m = 55; m < 67; m++)
                        {
                            if (incendent[j][m] == 1)
                            {
                                result_cost[i]+=mark_safe[m-55] ;

                            }
                        }
                    }

                }
            }
        }
        }
```

Algoritm2.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
```

```
namespace Course_Work.Models
{
    public class Algoritm2
    {
        public int Variable1 { get; set; }
        public int Variable2 { get; set; }

        public Course_Work.Controllers.Algoritm2Controller Algoritm2Controller
        {
            get
            {
                throw new System.NotImplementedException();
            }
            set
            {
            }
        }

        public static List<int> VersionFirst = new List<int>();
        public static List<int> VersionSecond = new List<int>();
        public static List<int> VersionThird = new List<int>();
        public static int amountVersionInFirstPD = 0;
        public static int amountVersionInSecondPD = 0;
        public static int sum2 = 0;

        public static void PodgrafThree()
        {

            double MinElemebtOfMatrix = Matrix.road_l[0][1];
            int FirstVi = 0;
            int SecondVi = 0;

            do
            {
            } while (amountVersionInFirstPD > Matrix.amount_ver);
            for (int i = 0; i < Matrix.amount_ver; i++)
            {
                for (int j = 0; j < Matrix.amount_ver; j++)
                {
                    if (MinElemebtOfMatrix > Matrix.road_l[i][j] && i != j)
                    {
                        MinElemebtOfMatrix = Matrix.road_l[i][j];
                        FirstVi = i + 1;
                        SecondVi = j + 1;
                    }
                }
            }


            VersionFirst.Add(FirstVi);
            VersionFirst.Add(SecondVi);
            int TempKOLverF = 2;
            while (amountVersionInFirstPD > TempKOLverF)
            {
                int current = SecondVi;
                int temp = SecondVi;
                double versionTemp = Matrix.road_l[SecondVi -
1][FirstValue(TempKOLverF)];
                for (int i = 0; i < Matrix.amount_ver; i++)
                {
                    if (versionTemp > Matrix.road_l[current - 1][i] && SimpleVersion(i +
1, TempKOLverF))
```

```
                {
                    versionTemp = Matrix.road_l[SecondVi - 1][i];
                    current = i + 1;
                    SecondVi = current;
                }
                if (i == Matrix.amount_ver-1 && temp == SecondVi)
                {
                    current = 0 + 1;
                    SecondVi = current;
                }
            }

            VersionFirst.Add(current);
            TempKOLverF++;
        }

        int TempKOLverS = 0;
            for (int j = 0; j < Matrix.amount_ver; j++)
            {
                bool test2 = true;
                for (int i = 0; i < TempKOLverF; i++)
                {
                    if ((j + 1) == VersionFirst[i])
                    {
                        test2 = false;
                    }
                }
                if (test2 && (amountVersionInSecondPD > TempKOLverS))
                {
                    VersionSecond.Add(j + 1);
                    TempKOLverS++;
                }
            }
        for (int j = 0; j < Matrix.amount_ver; j++)
        {
            bool test = true;
            for (int i = 0; i < TempKOLverF; i++)
            {
                if ((j + 1) == VersionFirst[i])
                {
                    test = false;
                }
            }
            for (int i = 0; i < TempKOLverS; i++)
            {
                if ((j + 1) == VersionSecond[i])
                {
                    test = false;
                }
            }
            if (test) VersionThird.Add(j + 1);

        }
    }
    public static int FirstValue(int Amount)
    {
        int temp = 0;
        for (int i = 0; i < Amount; i++)
        {
            for (int j = 0; j < Matrix.amount_ver; j++)
            {
```

```
                    if (VersionFirst[i] == j + 1)
                    {
                        continue;
                    }
                    else
                    {
                        temp = j;
                    }
                }
            }
            return temp;
        }

        public static bool SimpleVersion(int j, int Amount)
        {
            for (int i = 0; i < Amount; i++)
            {
                if (j == VersionFirst[i])
                {
                    return false;
                }
            }
            return true;
        }

    }
```

Matrix.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.IO;
namespace Course_Work.Models
{
    public class Matrix
    {
       public static int amount_ver = 0;
    public   static  int amount_dependency = 0;
    public static List<List<int>> incendent = new List<List<int>>();
        public static double[] important = new double[19];
        public static double[] problem = new double[38];
        public static double[] safe = new double[13];
        public static double[] cost = new double[13];
    public static List<List<double>> v = new List<List<double>>();
    public static List<List<double>> ves = new List<List<double>>();
     public static List<double> row = new List<double>();
     public static List<int> rowi = new List<int>();
     public static  List<List<double>> road_l = new List<List<double>>();
    public  static List<double> row_l = new List<double>();
    public static List<double> row_v = new List<double>();
   public static List<int> VersionFirst = new List<int>();
  public  static List<int> VersionSecond = new List<int>();
   public static int amountVersionInFirstPD = 0;

   public Course_Work.Controllers.MatrixController MatrixController
   {
       get
       {
           throw new System.NotImplementedException();
       }
```

```
set
     {
     }
 }

    public static void Incedent(int a, int b)
     { for (int i=0; i<amount_ver; i++)
         for (int j = 0; j < amount_ver; j++)
         {
             if (i == (a-1) && j==b-1)
             {
                 incendent[i][j] = 1;

                 incendent[j][i] = 1;
             }

         }
     }
    public static void Data(int a, double b)
    {
                if (a<=18)
                {
                    important[a-1]=b;
                }
                          if (a>18 && a<=55)
                {
                    problem[a-19]=b;
                }
                                      if (a>55)
                {
                   safe[a-56]=b;
                }
    }
    public static void Cost(int a, double b)
    {
        cost[a - 56] = b;
    }
    public static void values(int a, int b, double c)
    {
        for (int i = 0; i < amount_ver; i++)
            for (int j = 0; j < amount_ver; j++)
            {
                if (i == (a - 1) && j == b - 1)
                {
                    v[i][j] = c;

                    v[j][i] = c;
                }

            }
    }
    public static void zero_v()
    {
        for (int i = 0; i < amount_ver; i++)
        {
            row_v = new List<double>();
            for (int j = 0; j < amount_ver; j++)
                row_v.Add(0);
        }
    }
    public static void Matrix_v(int a, int b,double c)
    {
```

```
            for (int i = 0; i < amount_ver; i++)
                for (int j = 0; j < amount_ver; j++)
                {
                    if (i == (a - 1) && j == b - 1)


                    {
                        ves[i][j] = c;

                        ves[j][i] = c;
                    }

                }
        }

         public static void ReadFile()
       {
           using (var stream =
File.OpenText("C:/Users/User/Desktop/Course_Work/Course_Work/App_start/App_Data/graf.txt"
))
               {
                   string line;
                   int first = 0;
                   while ((line = stream.ReadLine()) != null)
                   {
                       var columns = line.Split(' ');

                       if (first ==0)
                       {
                           amount_ver = Int32.Parse(columns[0]);
                           first++;
                           amount_dependency = Int32.Parse(columns[1]);
                           for (int i=0; i<amount_ver; i++)
        {
            rowi = new List<int>();
            for (int j = 0; j < amount_ver; j++) rowi.Add(0);
            incendent.Add(rowi);

        }


                       }
                       else
                       {


                           Incedent(Int32.Parse(columns[0]),Int32.Parse(columns[1]));

                       }

                   }



               }
           }
        public static void ReadDataFile()
         {
             using (var stream =
File.OpenText("C:/Users/User/Desktop/Course_Work/Course_Work/App_start/App_Data/data.txt"
))
             {
                 string line;
```

```
                while ((line = stream.ReadLine()) != null)
                {
                    var columns = line.Split(' ');
                        Data(Int32.Parse(columns[0]), Double.Parse(columns[1]));

                }



            }
        }
       public static void ReadCostFile()
       {
           using (var stream =
File.OpenText("C:/Users/User/Desktop/Course_Work/Course_Work/App_start/App_Data/cost.txt"
))
           {
               string line;
               while ((line = stream.ReadLine()) != null)
               {
                   var columns = line.Split(' ');
                   Cost(Int32.Parse(columns[0]), Double.Parse(columns[1]));

               }
```

Sumroz.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.IO;

namespace Course_Work.Models
{
    public class Sumroz
    {
        public static int amount_ver = 0;
        public static List<List<double>> ves = new List<List<double>>();
        public static List<double> ver = new List<double>();
        public static List<double> row = new List<double>();
        public static double sum = 0;
        public static List<int> VersionFirst = new List<int>();
        public Course_Work.Controllers.SumrozController SumrozController
        {
            get
            {
                throw new System.NotImplementedException();
            }
            set
            {
            }
        }

        public static void Matrix_v(int a, int b, double c)
        {
            for (int i = 0; i <amount_ver; i++)
                for (int j = 0; j <amount_ver; j++)
                {
                    if (i == (a - 1) && j == b - 1)
                    {
                        ves[i][j] = c;
```

```
                    ves[j][i] = c;
                }

            }
        }
        public static void sumator()
        {
                    for (int i = 0; i <amount_ver; i++)
            for (int j = 0; j <amount_ver; j++)
            {
                ver[i] += ves[i][j];
            }
        }
        public static void zero_v()
        {
            for (int i = 0; i < amount_ver; i++)
                for (int j = 0; j < amount_ver; j++)
                {
                    ves[i][j] = 0;
                }
        }
        public static void zero_ver()
        {
            for (int i = 0; i < amount_ver; i++)
            {
                ver.Add(0);
            }
        }

        public static void PodgrafTwo()
        {
            double MaxElemebtOfMatrix = ver[0];
            int FirstVi = 0;
            for (int i = 0; i < 10; i++)
            {
                if (MaxElemebtOfMatrix < ver[i])
                {
                    MaxElemebtOfMatrix = ver[i];
                    FirstVi = i + 1;
                }


            }

            VersionFirst.Add(FirstVi);

            for (int i = 10; i < 20; i++)
            {
                if (ves[FirstVi - 1][i - 1] !=0)
                {
                    VersionFirst.Add(i + 1);
                }
            }

        }

        public static void ReadFile()
        {
            using (var stream =
File.OpenText("C:/Users/User/Desktop/Course_Work/Course_Work/App_start/App_Data/graf.txt"
))
            {
                string line;
```

```
                int first = 0;
                while ((line = stream.ReadLine()) != null)
                {
                    var columns = line.Split(' ');

                    if (first == 0)
                    {
                        amount_ver = Int32.Parse(columns[0]);
                        first++;
                        for (int i = 0; i < amount_ver; i++)
                        {
                            row = new List<double>();
                            for (int j = 0; j < amount_ver; j++) row.Add(0);
                            ves.Add(row);
                        } zero_v(); zero_ver();


                    }
                    else
                    {
                        Matrix_v(Int32.Parse(columns[0]), Int32.Parse(columns[1]),
Double.Parse(columns[2]));

                    }
```

Зміст

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
“КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМЕНІ ІГОРЯ СІКОРСЬКОГО”
Кафедра автоматизованих систем обробки інформації та управління

УЗГОДЖЕНО

Керівник проекту

_____ *Нестренко О.В.*
(підпис) (ініціали, прізвище)

“13” травня 2019 р.

ЗАТВЕРДЖУЮ

В.о. завідувача кафедри

_____ *О.А.Павлов*
(підпис) (ініціали, прізвище)

“14” травня 2019 р.

Графова модель безпеки ресурсів інформаційної системи

ПРОГРАМА ТА МЕТОДИКА ВИПРОБУВАНЬ

Шифр *ДП ІС-5102.1181-с.ПМВ*

на 10 сторінках

Київ – 2019 року

Зміст

1 ОБ'ЄКТ ВИПРОБУВАННЯ

1.1 Найменування програми

Повне найменування системи *«Графова модель безпеки ресурсів інформаційної системи».*

1.2 Область застосування

Програма розроблена для державних установ та великих компаній для підтримки безпеки ресурсів інформаційної системи та забезпечення захисту важливої інформації, а також визначення необхідності встановлення захисту для системи.

Результатом цього буде покращення рівня захисту інформації для важливих державних установ, а також це зменшить втрати, які могли б виникнути внаслідок здійснення загроз.

1.3 Умовне позначення програми

Умовне позначення програми ГМБ.

2 МЕТА ВИПРОБУВАНЬ

Мета приймальних випробувань полягає у підтвердженні відповідності проведених робіт затвердженому Технічному завданню.

3 ВИМОГИ ДО ПРОГРАМНОГО ПРОДУКТУ

3.1 Вимоги до функціональних характеристик

Система повинна виконувати наступні функції:

- створення переліку основних ресурсів системи та визначення їх цінності;
- створення переліку основних загроз для кожного типу ресурсів та визначення збитків, які виникнуть внаслідок спрацювання загрози, а також визначення ймовірності настання кожної загрози;
- створення переліку усіх наявних механізмів захисту для кожного типу загроз з оцінкою їх ефективності;
- формування основних підграфів системи;
- визначення доцільності використання наявних механізмів захисту на основі аналізу даних розрахунку кількісних оцінок.

3.1.1 Вимоги до складу виконуваних функцій

Система повинна містити наступні функції:

– аналіз вхідних даних;
– ведення моніторингу загроз;
- формування основних підграфів;
– розрахунок кількісних оцінок;
– формування звітності.

4 ВИМОГИ ДО ПРОГРАМНОЇ ДОКУМЕНТАЦІЇ

Програмна документація має бути розроблена згідно нормативних вимог і містити наступні документи:

- технічне завдання;
- керівництво користувача;
- пояснювальна записка.

5 СКЛАД І ПОРЯДОК ВИПРОБУВАНЬ

Приймальні випробування виконуються приймальною комісією у складі уповноважених осіб Замовника та Виконавця. Випробовування проводяться методом ручного тестування. По результатам оформлюється Акт здачі-приймання робіт. Приймальні випробування здійснюються на базі технічних засобів Замовника.

6 МЕТОДИ ВИПРОБУВАНЬ

Під час тестування була перевірена вся функціональність комплексу задач (КЗ). В таблицях 6.1-6.3 наведено перелік випробувань основних функціональних можливостей.

Таблиця 6.1 – Відображення матриці суміжності

Мета тесту	**Перевірка функції «Відображення матриці суміжності»**
Початковий стан КЗ	Відкрита головна сторінка програми
Схема проведення тесту	Натисну на вкладку «Матриця» та очікувати результат
Очікуваний результат	Відкрита сторінка з відображенням двох матриць суміжності. Перша матриця відображає зв’язок загроз та ресурсів, а друга загроз та механізмів захисту
Стан КЗ після проведення випробувань:	Відкрита сторінка «Матриця» з відображеною інформацією та можливістю переходити на інші вкладки меню

Таблиця 6.2 – Аналіз вхідних даних

Мета тесту	**Перевірка функції «Аналіз вхідних даних»**
Початковий стан КЗ	Відкрита сторінка «Матриця»
Схема проведення тесту	Натиснути на кнопку «Вхідні дані»

Продовження таблиці 6.2

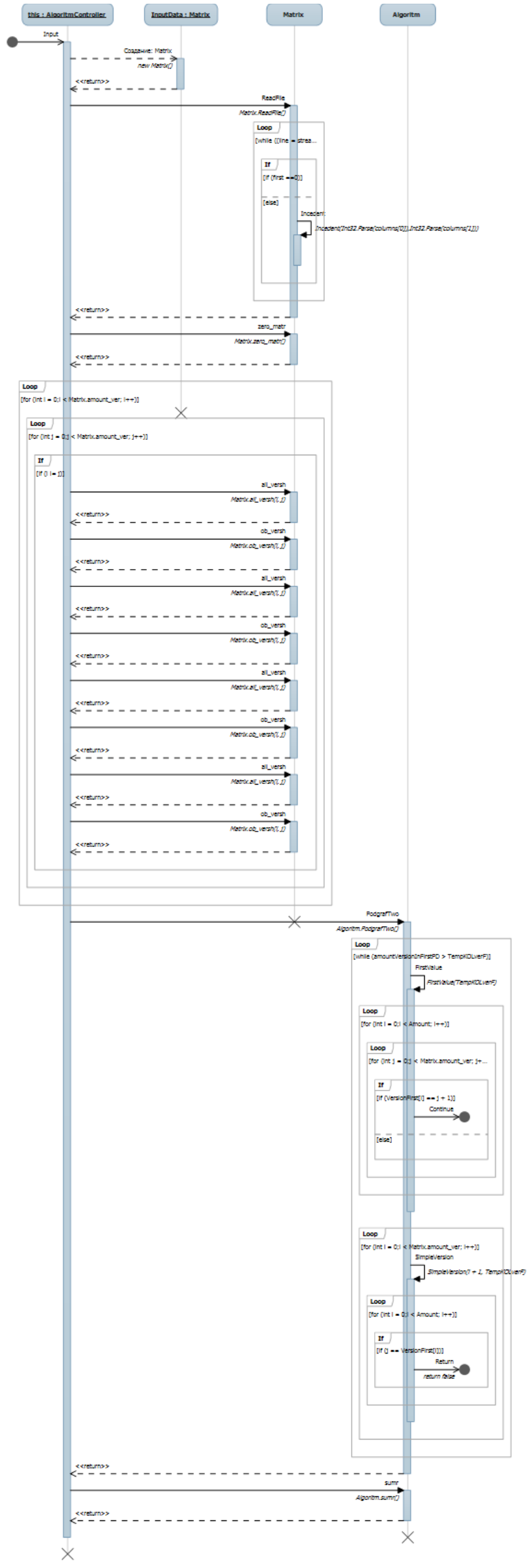
Мета тесту	**Перевірка функції «Аналіз вхідних даних»**
Очікуваний результат	Відкривається вкладка «Вхідні дані». На екрані відображається інформація у вигляді таблиць про цінність ресурсів, ймовірність виникнення загрози, вартість механізму захисту, а також ймовірність подолання загрози
Стан КЗ після проведення випробувань	Відкрита сторінка «Вхідні дані»

Таблиця 6.3 – Розрахунок кількісних оцінок

Мета тесту	**Перевірка функції «Розрахунок кількісних оцінок»**
Початковий стан КЗ	Відкрита сторінка «Вхідні дані»
Схема проведення тесту	Натиснути на вкладку «Розрахунки» та очікувати на результат
Очікуваний результат	Відкрита сторінка «Розрахунки». На екрані відображуються виділені підграфи, а також кількісні оцінки, на основі яких можна зробити висновок про доцільність встановлення механізму захисту

Продовження таблиці 6.3

Мета тесту	**Перевірка функції «Розрахунок кількісних оцінок»**
Стан КЗ після проведення випробувань	Відкрита сторінка «Розрахунки»



					ДП ІС-5102.1181-с.ССП	
Зм.	Арк.	№ докум.	Підп.	Дата		
Розроб.		Асламова М.С.			Схема структурна послідовностей	Лит. / Маса / Масштаб
Перев.		Нестеренко О.В.				
Т. Кон.						Аркуш 1 / Аркушів 1
					Графова модель безпеки ресурсів інформаційної системи	КПІ ім. Ігоря Сікорського кафедра АСОІУ гр. ІС-51
Н. Кон.		Тєлишева Т.О.				
Затв.		Нестеренко О.В.				

					ДП ІС-5102.1181-с.ССР		
Зм.	Арк.	№ документа	Підпис	Дата	*Схема структурна розгортання*	Літера	Маса / Масштаб
Розробив		Асламова М.С.					
Перевірив		Нестеренко О.В.				Аркуш 1	Аркушів 1
Т. кон.					*Графова модель безпеки ресурсів інформаційної системи*	*КПІ ім. Ігоря Сікорського кафедра АСОІУ гр. ІС-51*	
Н. кон.		Тєлишева Т.О.					
Затвердив		Нестеренко О.В.					

					ДП ІС-5102.1181-с.ССК		
Зм.	Арк.	№ документа	Підпис	Дата	Схема структурна класів програмного забезпечення	Літера	Маса / Масштаб
Розробив		Асламова М.С.					
						Аркуш 1	Аркушів 1
Перевірив		Нестеренко О.В.					
Т. кон.					Графова модель безпеки ресурсів інформаційної системи	КПІ ім. Ігоря Сікорського кафедра АСОІУ гр. ІС-51	
Н. кон.		Тєлишева Т.О.					
Затвердив		Нестеренко О.В.					

					ДП ІС-5102.1181-с.ССВ			
						Літера	Маса	Масштаб
Зм.	Арк.	№ документа	Підпис	Дата	Схема структурна варіантів використань			
Розробив		Асламова М.С.						
						Аркуш 1		Аркушів 1
Перевірив		Нестеренко О.В.						
Т. кон.					Графова модель безпеки ресурсів інформаційної системи	КПІ ім. Ігоря Сікорського кафедра АСОІУ гр. ІС-51		
Н. кон.		Тєлишева Т.О.						
Затвердив		Нестеренко О.В.						

Матриця, що показує зв'язок ресурсів та загроз

1 - є загроза для даного ресурсу, 0 - немає загрози

	Загроза 1	Загроза 2	Загроза 3	Загроза 4	Загроза 5	Загроза 6	Загроза 7	Загроза 8	Загроза 9	Загроза 10	Загроза 11	Загроза 12	Загроза 13	Загроза 14	Загроза 15	Загроза 16	Загроза 17	Загроза 18	Загроза 19	Загроза 20	Загроза 21	Загроза 22	Загроза 23	Загроза 24	Загроза 25	Загроза 26	Загроза 27	Загроза 28	Загроза 29	Загроза 30	Загроза 31	Загроза 32	Загроза 33	Загроза 34	Загроза 35	Загроза 36	Загроза 37
Ресурс 1	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
Ресурс 2	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0
Ресурс 3	0	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
Ресурс 4	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Ресурс 5	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Ресурс 6	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Ресурс 7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Ресурс 8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Ресурс 9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Ресурс 10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	1	0	1
Ресурс 11	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
Ресурс 12	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
Ресурс 13	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	1	0	0	0	0	0
Ресурс 14	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0
Ресурс 15	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	1	0	0
Ресурс 16	0	0	0	0	1	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Ресурс 17	0	0	1	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Ресурс 18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	1	0	0	0	0	0	0	0	0

Матриця, що показує зв'язок загроз та систем захисту

1 - СЗ спрацьовує, 0 - СЗ не спрацьовує

	СЗ 1	СЗ 2	СЗ 3	СЗ 4	СЗ 5	СЗ 6	СЗ 7	СЗ 8	СЗ 9	СЗ 10	СЗ 11	СЗ 12
Загроза 1	1	0	0	0	0	0	0	0	0	0	0	0
Загроза 2	0	0	1	0	0	0	0	0	0	0	0	0
Загроза 3	1	0	0	0	0	0	0	0	0	0	0	0
Загроза 4	0	0	0	0	0	0	0	0	0	0	0	1
Загроза 5	0	0	0	0	0	0	0	0	0	1	0	0
Загроза 6	0	0	0	0	0	0	0	0	0	0	1	0
Загроза 7	0	1	0	0	0	0	0	0	0	0	0	0
Загроза 8	0	0	0	0	1	0	0	0	0	0	0	0
Загроза 9	0	0	0	0	1	0	0	0	0	0	0	0
Загроза 10	1	0	0	0	0	0	0	0	0	0	0	0
Загроза 11	0	0	0	0	0	1	0	0	0	0	0	0
Загроза 12	0	0	0	0	0	0	1	0	0	0	0	0
Загроза 13	0	0	0	0	0	0	0	1	0	0	0	0
Загроза 14	0	0	0	0	0	0	0	0	0	0	0	0
Загроза 15	0	0	0	0	0	0	0	0	1	0	0	0
Загроза 16	0	0	0	0	0	0	0	0	0	0	1	0
Загроза 17	0	0	0	0	0	0	0	0	1	0	0	0
Загроза 18	0	0	0	0	0	0	1	0	0	0	0	0

Кількість загроз для кожного ресурсу

Ресурс 1	Ресурс 2	Ресурс 3	Ресурс 4	Ресурс 5	Ресурс 6	Ресурс 7	Ресурс 8	Ресурс 9	Ресурс 10	Ресурс 11	Ресурс 12	Ресурс 13	Ресурс 14	Ресурс 15	Ресурс 16	Ресурс 17	Ресурс 18
4	4	2	3	4	3	3	2	1	5	4	4	4	3	5	4	3	3

Утворені підграфи

Підграф 1	15	23	24	42	44	53	1	11	16	7	10	18	25	35	45	54	26	27	28	37	40	41	43	55	47												
Підграф 2	10	41	42	43	53	55	7	15	37	40	23	24	44																								
Підграф 3	16	23	26	27	28	1	11	15	3	4	25	35	45	24	54	42	44	53	22	29	36																
Підграф 4	13	48	50	51	52	2	21																														
Підграф 5	12	36	37	38	54	4	7	8	11	28	29	40	42	39	23	24	25																				
Підграф 6	11	23	24	25	54	1	15	16	12	35	45	42	44	53	26	27	28	30	37	38																	
Підграф 7	5	30	32	33	35	1	9	23	25	45																											
Підграф 8	2	27	50	51	52	17	13	22	29	48																											

Цінність ресурсів

Ресурс 1	Ресурс 2	Ресурс 3	Ресурс 4	Ресурс 5	Ресурс 6	Ресурс 7	Ресурс 8	Ресурс 9	Ресурс 10	Ресурс 11	Ресурс 12	Ресурс 13	Ресурс 14	Ресурс 15	Ресурс 16	Ресурс 17	Ресурс 18
100	100	50	100	75	55	70	80	90	100	30	100	10	45	70	80	90	100

Вартість механізму захисту

МЗ 1	МЗ 2	МЗ 3	МЗ 4	МЗ 5	МЗ 6	МЗ 7	МЗ 8	МЗ 9	МЗ 10	МЗ 11	МЗ 12
95	85	75	60	45	95	75	65	60	70	45	40

Ймовірність виникнення загрози

Загроза 1	Загроза 2	Загроза 3	Загроза 4	Загроза 5	Загроза 6	Загроза 7	Загроза 8	Загроза 9	Загроза 10	Загроза 11	Загроза 12	Загроза 13	Загроза 14	Загроза 15	Загроза 16	Загроза 17	Загроза 18	Загроза 19	Загроза 20	Загроза 21	Загроза 22	Загроза 23	Загроза 24	Загроза 25	Загроза 26	Загроза 27	Загроза 28	Загроза 29	Загроза 30	Загроза 31	Загроза 32	Загроза 33	Загроза 34	Загроза 35	Загроза 36	Загроза 37
0,2	0,4	0,7	0,3	0,25	0,5	0,2	0,4	0,45	0,4	0,35	0,8	0,9	0,35	0,45	0,5	0,55	0,45	0,35	0,4	0,45	0,3	0,4	0,95	0,8	0,35	0,4	0,35	0,4	0,8	0,75	0,7	0,45	0,35	0,35	0,3	0,45

Ймовірність усунення загрози

СЗ 1	СЗ 2	СЗ 3	СЗ 4	СЗ 5	СЗ 6	СЗ 7	СЗ 8	СЗ 9	СЗ 10	СЗ 11	СЗ 12
0,9	0,75	0,65	0,5	0,75	0,85	0,9	0,55	0,99	0,75	0,85	0,75

Втрати для кожного ресурсу без МЗ

Ресурс 1	Ресурс 2	Ресурс 3	Ресурс 4	Ресурс 5	Ресурс 6	Ресурс 7	Ресурс 8	Ресурс 9	Ресурс 10	Ресурс 11	Ресурс 12	Ресурс 13	Ресурс 14	Ресурс 15	Ресурс 16	Ресурс 17	Ресурс 18
55	70	22,5	45	60	49,5	66,5	36	49,5	95	15	45	8	18	66,5	36	63	40

Втрати для кожного ресурсу при застосуванні МЗ

Ресурс 1	Ресурс 2	Ресурс 3	Ресурс 4	Ресурс 5	Ресурс 6	Ресурс 7	Ресурс 8	Ресурс 9	Ресурс 10	Ресурс 11	Ресурс 12	Ресурс 13	Ресурс 14	Ресурс 15	Ресурс 16	Ресурс 17	Ресурс 18
13,75	17,5	5,625	11,25	15	12,375	16,625	9	12,375	23,75	3,75	11,25	2	4,5	16,625	9	15,75	10

Кількість загроз, на які впливає МЗ

МЗ 1	МЗ 2	МЗ 3	МЗ 4	МЗ 5	МЗ 6	МЗ 7	МЗ 8	МЗ 9	МЗ 10	МЗ 11	МЗ 12
3	3	4	3	5	1	4	2	3	2	3	1

Оцінка МЗ

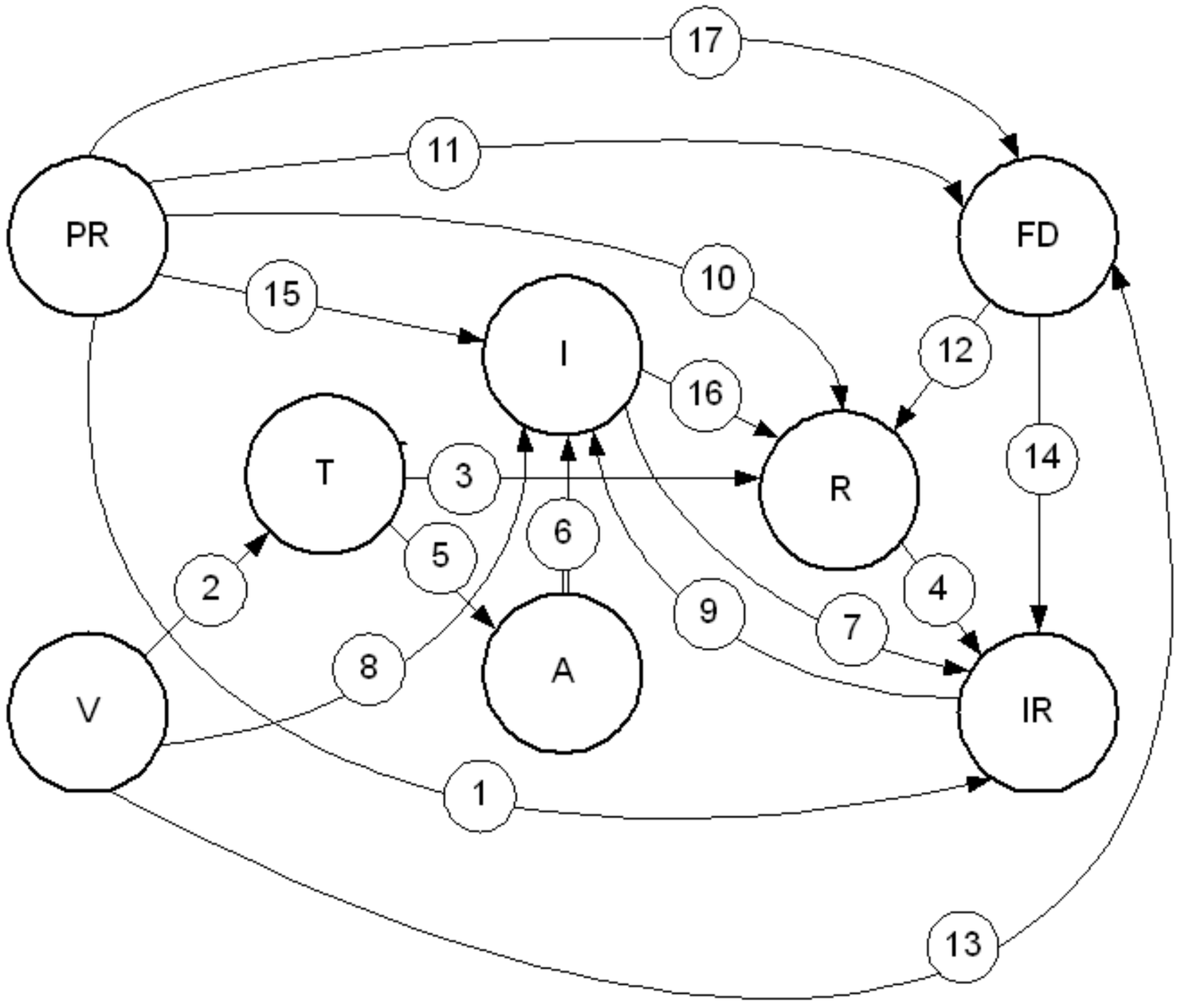
МЗ 1	МЗ 2	МЗ 3	МЗ 4	МЗ 5	МЗ 6	МЗ 7	МЗ 8	МЗ 9	МЗ 10	МЗ 11	МЗ 12
19	28,33	18,75	20	9	95	18,75	32,5	20	35	15	40

Вартість встановлення МЗ для кожного об'єкту

Ресурс1	Ресурс2	Ресурс3	Ресурс4	Ресурс5	Ресурс6	Ресурс7	Ресурс8	Ресурс9	Ресурс10	Ресурс11	Ресурс12	Ресурс13	Ресурс14	Ресурс15	Ресурс16	Ресурс17	Ресурс18
92,33	89	49	132,75	58,75	67,5	56,5	48,33	20	93,58	97,33	85,08	88,75	38	107,5	72	154	38

					ДП ІС-5102.1181-с.КЕ		
Зм.	Арк.	№ документа	Підпис	Дата	Креслення вигляду екранних форм	Літера	Маса / Масштаб
Розробив		Асламова М.С.					
Перевірив		Нестеренко О.В.				Аркуш 1	Аркушів 1
Т. кон.							
Н. кон.		Телишева Т.О.			Графова модель безпеки ресурсів інформаційної системи	КПІ ім. Ігоря Сікорського кафедра АСОІУ гр. ІС-51	
Затвердив		Нестеренко О.В.					

Рішення з математичного забезпечення



Демонстраційний плакат до дипломного проекту

«Графова модель безпеки ресурсів інформаційної системи»

Виконала студентка гр.ІС-51 Асламова М.С.

Керівник ДП Нестеренко О.В.